
BACHELORARBEIT

Frau
Jana Bedlewski

**Konzeption und Entwicklung
eines Mitfahrdienstes**

Mittweida, 2010

BACHELORARBEIT

Konzeption und Entwicklung eines Mitfahrdienstes

Autor:

Frau Jana Bedlewski

Studiengang:

Informationstechnologie

Seminargruppe:

CT04w1

Erstprüfer:

Prof. Dr.-Ing. habil. Lutz Winkler

Zweitprüfer:

M.Sc. Rico Thomanek

Einreichung:

Mittweida, 30.11.2010

Verteidigung/Bewertung:

Mittweida, 2010

Bibliografische Angaben:

Bedlewski, Jana:

Konzeption und Entwicklung eines Mitfahrdienstes - 2010 - VI, 155, IV S.
Mittweida, Hochschule Mittweida (FH), University of Applied Sciences,
Fakultät Elektro- und Informationstechnik, Bachelorarbeit, 2010

Referat:

Das Studentenportal als Anlaufpunkt für verschiedenste Dienstleistungen aus dem studentischen Alltag soll mit weiteren sinnvollen Angeboten ausgebaut werden.

Diese Bachelorarbeit befasst sich mit dem Entwurf und der praktischen Umsetzung eines Dienstes zur Vermittlung von Mitfahrgelegenheiten. Dabei werden zunächst bereits bestehende Web-basierte Serviceangebote dieser Art analysiert, um daraus die nötigen Grundbestandteile herleiten zu können. Die Voraussetzungen des Studentenportals und die Analyse einer im Jahr 2003 realisierten Variante des Mitfahrdienstes werden darüber hinaus hinzugezogen, um anschließend einen konkreten Dienstentwurf zu erarbeiten. Dieser erläutert alle zur Anwendung vorgesehenen Bestandteile und Aspekte. Die Beschreibung der programmtechnischen Umsetzung bietet letztendlich einen genauen Einblick in deren konkrete Details und ist gleichzeitig eine wichtige Voraussetzung bzw. Grundlage einer eventuellen späteren Weiterentwicklung. Eine abschließende Auswahl von Erweiterungs- bzw. Verbesserungsvorschlägen stellt dafür einen möglichen Ausgangspunkt bereit.

Inhaltverzeichnis

Inhaltverzeichnis	I
Abbildungsverzeichnis	III
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
0 Einleitung	1
0.1 Motivation und Zielstellung.....	1
0.2 Kapitelübersicht.....	2
1 Grundlagen	3
1.1 Analyse der Konzepte vorhandener Mitfahrdienste im Internet.....	3
1.1.1 Mitfahrgelegenheit	3
1.1.2 Drive2Day	5
1.1.3 raumobil	8
1.1.4 Mitfahrzentrale	10
1.1.5 MFZ	13
1.1.6 Liftpool	15
1.1.7 Detailauflistung	17
1.2 Webbasierte Umfrage	24
1.3 Allgemeine Anforderungen an einen Mitfahrdienst	27
2 Dienstentwurf	31
2.1 Umgebung des Dienstes – das Studentenportal	31
2.2 Projektstand bei Übernahme.....	34
2.2.1 Kritik.....	34
2.2.2 Fazit	36
2.3 Spezifikation des Dienstes	38
2.3.1 Grundkonzept und Struktur	38
2.3.2 Eintragung/Verwaltung von Inseraten	39
2.3.3 Suchfunktion.....	45
2.3.4 Buchen.....	47
2.3.5 Bewerten.....	49
2.3.6 Detailinformationen zu den Nutzern des Dienstes	51
2.3.7 Kontaktaufnahme/Benachrichtigung	53
3 Programmierumgebung	55
3.1 Übersichtsdiagramm Programmseiten / Bibliotheken / Klassen	55
3.2 Verwendete Bibliotheken / Klassen.....	56
3.3 Datenbankstruktur.....	68

4	Programmtechnische Umsetzung	76
4.1	Anmerkungen zur objektbasierten Programmierung	76
4.2	Anmerkungen zur Programmierung der Übersichtsseiten	78
4.3	Anmerkungen zur Verwendung von JavaScript	78
4.4	Startseite des Dienstes	79
4.5	Seite zur Erstellung eines Angebots / Gesuchs	82
4.6	Seiten für die Übersicht eingetragener Angebote / Gesuche	92
4.7	Seite zur Darstellung der Details der Angebote und Gesuche	95
4.8	Seite zum Löschen eingetragener Angebote und Gesuche	97
4.9	Seite zum Löschen einer Eintragung in der Favoritenliste	99
4.10	Seiten zur Durchsuchung eingetragener Angebote / Gesuche	101
4.11	Seiten zur Verarbeitung und Darstellung der Suchergebnisse	104
4.12	Seite zur Buchung eines Angebotes / Gesuches	110
4.13	Seiten für die Übersicht vorgenommener Buchungen	120
4.14	Seite zur Darstellung der Details einer Buchung	124
4.15	Seite zum Löschen einer Buchung	125
4.16	Seite zur Bewertung eines Fahrers / Mitfahrers	127
4.17	Seiten für die Übersicht vorgenommener bzw. ausstehender Bewertungen	132
4.18	Seite zur Darstellung der Details einer Bewertung	137
4.19	Seite zum Verschicken einer Nachricht	139
4.20	Seite zur Eintragung des persönlichen Profils	141
4.21	Seite zur Eintragung des Fahrzeugprofils	144
4.22	Seite zur Darstellung der personenbezogenen Informationen	146
4.23	Seite zur Durchsuchung eingetragener Nutzerprofile	149
4.24	Seiten zur Verarbeitung und Darstellung der Suchergebnisse	150
5	Zusammenfassung und Ausblick	152
5.1	Ergebnisse	152
5.2	Erweiterungsvorschläge	153
Anhang		I
Anhang A		I
Quelltexte		I
Anhang B		I
Kurzreferenz		I
Literaturverzeichnis		III
Ausgewertete Mitfahrdienste		IV
Selbstständigkeitserklärung		

Abbildungsverzeichnis

ABBILDUNG 1-1 SCREENSHOT STARTSEITE WWW.MITFAHRGELEGENHEIT.DE	3
ABBILDUNG 1-2 SCREENSHOT STARTSEITE WWW.DRIVE2DAY.DE	5
ABBILDUNG 1-3 SCREENSHOT STARTSEITE WWW.RAUMOBIL.DE	8
ABBILDUNG 1-4 SCREENSHOT STARTSEITE WWW.MITFAHRZENTRALE.DE	10
ABBILDUNG 1-5 SCREENSHOT STARTSEITE WWW.MFZ.DE	13
ABBILDUNG 1-6 SCREENSHOT PROFISUCHE WWW.LIFTPOOL.DE	15
ABBILDUNG 1-7 AUSGEWÄHLTE UMFRAGEERGEBNISSE - ALLGEMEIN (MICROSOFT EXCEL 2003)	24
ABBILDUNG 1-8 AUSGEWÄHLTE UMFRAGEERGEBNISSE - FAHRERSICHT (MICROSOFT EXCEL 2003)	25
ABBILDUNG 1-9 AUSGEWÄHLTE UMFRAGEERGEBNISSE - MITFAHRERSICHT (MICROSOFT EXCEL 2003)	26
ABBILDUNG 2-1 SCREENSHOT STARTSEITE STUDENTENPORTAL DER HOCHSCHULWEBSITE	31
ABBILDUNG 3-1 ÜBERSICHT PROGRAMMSEITEN / BIBLIOTHEKEN / KLASSEN (MICROSOFT VISIO 2007)	55
ABBILDUNG 3-2 SCREENSHOT TABELLENVERKNÜPFUNGEN (MICROSOFT ACCESS 2007)	68
ABBILDUNG 3-3 SCREENSHOT TABELLE DES PERSÖNLICHEN PROFILS	70
ABBILDUNG 3-4 SCREENSHOT TABELLE DES FAHRZEUGPROFILS	71
ABBILDUNG 3-5 SCREENSHOT TABELLE DER INSERATE	72
ABBILDUNG 3-6 SCREENSHOT TABELLE DER VORGEMERKTEN INSERATE	73
ABBILDUNG 3-7 SCREENSHOT TABELLE DER VORGENOMMENEN BUCHUNGEN	73
ABBILDUNG 3-8 SCREENSHOT TABELLE DER TEMPORÄREN BEWERTUNGSDATENSÄTZE	74
ABBILDUNG 3-9 SCREENSHOT TABELLE DER ENDGÜLTIGEN BEWERTUNGSDATENSÄTZE	75
ABBILDUNG 4-1 SCREENSHOT STARTSEITE	80
ABBILDUNG 4-2 SCREENSHOT INSERATSFORMULAR	85
ABBILDUNG 4-3 SCREENSHOT INSERATSÜBERSICHT	94
ABBILDUNG 4-4 SCREENSHOT INSERATSDetails	95
ABBILDUNG 4-5 SCREENSHOT LÖSCHEN EINES INSERATS	97
ABBILDUNG 4-6 SCREENSHOT LÖSCHEN EINES EINTRAGS DER FAVORITENLISTE	99
ABBILDUNG 4-7 SCREENSHOT DETAILSUCHE	102
ABBILDUNG 4-8 SCREENSHOT SUCHERGEBNISSE	105
ABBILDUNG 4-9 SCREENSHOT BUCHUNGSFORMULAR	112
ABBILDUNG 4-10 SCREENSHOT BUCHUNGSÜBERSICHT (HAUPTSEITE)	121
ABBILDUNG 4-11 SCREENSHOT BUCHUNGSÜBERSICHT (FREMDE ANGEBOTE)	122

ABBILDUNG 4-12 SCREENSHOT BUCHUNGSDetails	124
ABBILDUNG 4-13 SCREENSHOT LÖSCHEN EINER BUCHUNG	125
ABBILDUNG 4-14 SCREENSHOT BEWERTUNGSFORMULAR.....	128
ABBILDUNG 4-15 SCREENSHOT BEWERTUNGSÜBERSICHT (HAUPTSEITE)	134
ABBILDUNG 4-16 SCREENSHOT BEWERTUNGSÜBERSICHT (FREIGEgeben).....	134
ABBILDUNG 4-17 SCREENSHOT BEWERTUNGSÜBERSICHT (ANGEgeben)	135
ABBILDUNG 4-18 SCREENSHOT BEWERTUNGSDETAILS.....	137
ABBILDUNG 4-19 SCREENSHOT MAILFORMULAR	139
ABBILDUNG 4-20 SCREENSHOT FORMULAR PERSÖNLICHES PROFIL	142
ABBILDUNG 4-21 SCREENSHOT FORMULAR FAHRZEUGPROFIL.....	145
ABBILDUNG 4-22 SCREENSHOT PROFILDETAILS	147
ABBILDUNG 4-23 SCREENSHOT PERSONENSUCHE	149
ABBILDUNG 4-24 SCREENSHOT SUCHERGEBNISSE PERSONENSUCHE.....	150

Tabellenverzeichnis

TABELLE 1-1 VERGLEICH VORHANDENER MITFAHRDIENSTE - ALLGEMEINE MERKMALE	18
TABELLE 1-2 VERGLEICH VORHANDENER MITFAHRDIENSTE – SUCHE	19
TABELLE 1-3 VERGLEICH VORHANDENER MITFAHRDIENSTE - FAHRT(EN) EINTRAGEN	20
TABELLE 1-4 VERGLEICH VORHANDENER MITFAHRDIENSTE – BUCHUNGSSYSTEM	21
TABELLE 1-5 VERGLEICH VORHANDENER MITFAHRDIENSTE – BEWERTUNGSSYSTEM	21
TABELLE 1-6 VERGLEICH VORHANDENER MITFAHRDIENSTE – NUTZERPROFIL.....	22
TABELLE 1-7 VERGLEICH VORHANDENER MITFAHRDIENSTE – FAHRZEUGPROFIL.....	22
TABELLE 1-8 VERGLEICH VORHANDENER MITFAHRDIENSTE - VERWALTUNG	23

Abkürzungsverzeichnis

ASP	Active Server Pages
CD-ROM	Compact Disc Read-Only Memory
CSS	Cascading Style Sheets
E-Mail	electronic mail
HTML	Hypertext Markup Language
ID	Identifier (dt.: Identifikationsbezeichnung)
PHP	Hypertext Preprocessor
SQL	Structured Query Language
WWW	World Wide Web
WYSIWYG	„What You See Is What You Get“

0 Einleitung

0.1 *Motivation und Zielstellung*

Schon seit Herbst 2001 gab es erste Überlegungen zur Implementierung eines Studentenportals in das Intranet der Hochschule Mittweida. Grund hierfür war u.a. der bisher ungenügende Umfang an Serviceangeboten speziell für die Bedürfnisse der Studenten. Deshalb wurde als Erweiterung zu den bisher schon existierenden Intranetdiensten, welche eher allgemeine Angelegenheiten der Hochschule abdecken, ein Studentenportal als zentrale Anlaufstelle für Dienste rund um das studentische Leben konzipiert. Sowohl Dienste welche die internen Abläufe der Hochschule betreffen (bspw. Stundenplan, Noteneinsicht, Einschreibungen etc.), als auch Dienste für das private Leben (wie etwa ein Wohnungsmarkt) sind hier versammelt und schon fertig integriert. Jedoch bietet sich auch noch Raum für Erweiterungen und einige Bereiche befinden sich teilweise noch in der Entwicklung, und dazu zählt auch die Ausarbeitung eines Mitfahrdienstes.

Für Studenten ist die finanzielle Lage meist sowieso schon angespannt, und die ständig steigenden Lebenshaltungskosten, sowie Benzinpreise und Unterhaltskosten eines PKW sind auf Dauer eine große Belastung. Und hier setzt nun das Thema der Bachelorarbeit, die Entwicklung und Programmierung eines Mitfahrdienstes, an, um durch die Möglichkeit der einfachen Vermittlung von Fahrgelegenheiten den Transportaspekt betreffend eine Option der Abhilfe anzubieten. Auswärts wohnende Studenten pendeln sowieso täglich zur Hochschule, und wer zeitweise am Ort wohnt, fährt fast regelmäßig wöchentlich nach Hause. Somit gibt es zusätzlich zum finanziellen Aspekt mit der großen Anzahl an potentiellen Nutzern bzw. Interessenten einen weiteren Grund für den Nutzen eines solchen Dienstes. Weiterhin darf natürlich die günstige Auswirkung auf den Faktor Umwelt durch die optimierte Fahrzeugauslastung nicht außer Acht gelassen werden. Da heutzutage fast jeder Student zu Hause einen Internetanschluss besitzt, oder zumindest die Zugangsmöglichkeiten in der Hochschule nutzen kann, sollte dieser Dienst auch von allen Studenten jederzeit erreichbar und damit nutzbar sein.

Diese Arbeit soll nun dazu dienen einen theoretischen Entwurf eines Mitfahrdienstes für das Studentenportal des Intranets der Hochschule Mittweida zu erarbeiten, um anschließend daraus eine praktische, programmtechnische Umsetzung zu realisieren. Aus dem Jahr 2003 existiert bereits eine sehr ähnliche Arbeit¹, jedoch kam dieser Dienst damals noch nicht zum Einsatz, und hätte nun an die inzwischen vorherrschenden Umgebungskriterien angepasst und funktionsfähig gemacht werden müssen. Die erforderlichen Änderungen waren jedoch von einem solchen Umfang, der es sinnvoller erscheinen lies, einen kompletten Neuaufbau anzustreben. Dies hatte natürlich zur Folge, dass auch die theoretischen Betrachtungen zur Konzeption eines Mitfahrdienstes auf den aktuellen Stand gebracht werden mussten.

¹ Sandro Gerstenberger: Konzeption und Implementierung eines Mitfahrdienstes, 2003

0.2 Kapitelübersicht

Kapitel 0 soll an das Thema der Bachelorarbeit heranzuführen und einen kurzen Überblick darüber geben, welche Inhalte in den einzelnen Kapiteln der Bachelorarbeit erörtert werden.

Kapitel 1 befasst sich zunächst einmal mit verschiedenen im Internet angebotenen Seiten, welche einen Mitfahrdienst realisieren. Diese werden auf der Basis verschiedenster Kriterien verglichen und entsprechend ausgewertet. Zusätzlich dazu werden die gewonnenen Erkenntnisse aus der im Kreise von Studenten der Hochschule erarbeiteten Umfrage aus der Vorarbeit herangezogen um am Ende des Kapitels erste allgemeine Anforderungen für die Realisierung eines Mitfahrdienstes für das Studentenportal der Hochschule Mittweida zu formulieren.

Kapitel 2 beschäftigt sich mit der theoretischen Konzeption des Mitfahrdienstes. Dabei wird zunächst auf die Umgebung des Studentenportals eingegangen, bevor im zweiten Teil die Vorarbeit von Sandro Gerstenberger kurz analysiert wird. Zum Schluss werden dann die einzelnen Funktionalitäten und Teilbereiche des Dienstes aufgezeigt und erklärt.

Kapitel 3 stellt wichtige Umgebungsparameter der Programmierung vor. Dazu wird ein Überblick der Zusammenhänge verwendeter Bibliotheken und Seiten gegeben, die genutzten Bibliotheken werden im Bezug auf ihre Verwendung im Dienst beschrieben und die zu Grunde liegende Datenbankstruktur mit ihren einzelnen Tabellen dargelegt.

Kapitel 4 behandelt die konkrete praktische Umsetzung des Mitfahrdienstes. Die einzelnen entwickelten Seiten werden im Bezug auf ihre Funktionalität, verwendete Parameter etc. im Detail erläutert.

Kapitel 5 liefert ein kurzes Fazit zum erreichten Stand der Arbeit und unterbreitet Vorschläge für mögliche Erweiterungen des Dienstes.

1 Grundlagen

1.1 Analyse der Konzepte vorhandener Mitfahrdienste im Internet

Das Konzept „Mitfahrdienst“ wurde im Internet bereits auf einigen verschiedenen Websiten in jeweils unterschiedlicher Ausführung umgesetzt. Deshalb ist es sinnvoll, einige derartige Seiten herauszugreifen, um diese auf Art und Weise ihres zu Grunde liegenden Entwurfs hin zu analysieren. In diesem Abschnitt sollen deshalb sechs dieser, im deutschen Raum genutzten, Webauftritte vorgestellt, d.h. deren Konzepte betrachtet, sowie Vor- und Nachteile herausgearbeitet werden. Im Anschluss werden die detaillierten Ergebnisse, nach Funktionsbereichen getrennt, übersichtlich in einer Tabelle zusammengefasst.

1.1.1 Mitfahrgelegenheit

- Screenshot:

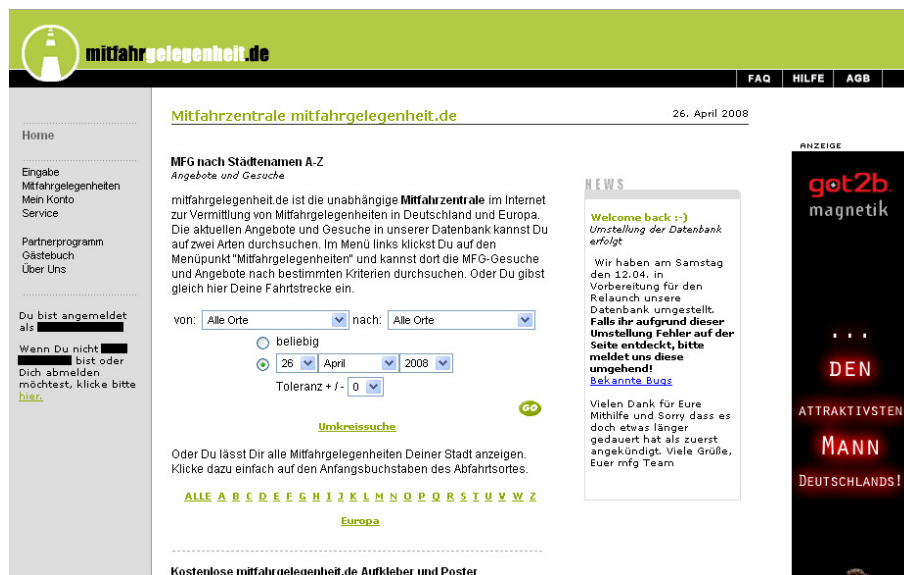


Abbildung 1-1 Screenshot Startseite www.mitfahrgelegenheit.de

- Funktionsweise:

Hierbei handelt es sich um einen nicht kommerziellen, unabhängigen Dienst, welcher 2001 gegründet wurde und unter **www.mitfahrgelegenheit.de** zu erreichen ist. Diese Seite überzeugt durch ihre relativ einfache Handhabung und einem schlichten, aber effektiven Design. Die große Anzahl an eingetragenen Inseraten lässt außerdem auf eine hohe Akzeptanz des Dienstes durch dessen Nutzer schließen. Das Grundprinzip ist denkbar einfach – **Angebote oder Gesuche** einer Mitfahrgelegenheit werden **in die Datenbank eingetragen**, und interessierte Nutzer können über eine Suchfunktion auf die Informationen zugreifen. Die Eingabe erfolgt in vier logischen und überschaubaren Schritten: Art des Inserats (Angebot oder Gesuch), allgemeine Häufigkeit (einmalig oder regelmäßig), Land (Deutschland oder Europa) und konkrete Details des Inserats. Die Detaildaten setzen sich aus folgenden Pflichteingaben zusammen: Datum, Uhrzeit, Star-

tort, Zielort, sowie Anzahl der verfügbaren oder gesuchten Plätze. Bei regelmäßigen Inseraten muss außerdem die konkrete Frequenz gewählt werden – täglich, werktäglich, wöchentlich (mit konkreter Angabe der einzelnen Wochentage), und bei Auslandsfahrten ist bei den einzelnen Streckenpunkten das Land zu wählen. Optional gibt es für den Nutzer noch die Möglichkeit der Angabe von Zwischenstationen, zusätzlichen Bemerkungen, sowie (nur bei Angeboten) dem Kfz-Kennzeichen und der Kennzeichnung als Raucher. Eine ungefähre Preisangabe ist hier nicht vorgesehen.

Bei der Auswahl der Orte ist auffallend, dass hier keine eigene Angabe gemacht werden kann; das heißt, es können nur vorgegebene Orte aus einer Liste ausgewählt werden. Dies hat natürlich den Nachteil, dass in den meisten Fällen keine kleineren Orte verfügbar sind, und dann der nächst größere genommen werden müsste. In so einem Fall erweist sich aber das Bemerkungsfeld als günstige Möglichkeit um für entsprechende Mitteilungen genutzt zu werden. Ergänzungen der Liste sind nur durch eine Mitteilung an den Betreiber möglich (über das Gästebuch), werden aber nicht garantiert.

Vorteilhaft erweist sich bei der Eingabe die **Prüfung auf fehlende oder inkorrekte Daten**. Alle eingestellten Inserate kann der jeweilige Nutzer **auf einer übersichtlichen Seite verwalten**, d.h. bearbeiten, löschen, oder kopieren (ein neues Inserat aus den vorliegenden Daten erstellen).

Die **Suchfunktion** ist recht einfach gehalten, was aber kein Nachteil sein soll, denn für diese Seite ist sie völlig ausreichend. Gesucht werden kann nach Art (Angebot, Gesuch oder beides), Ort (Startort und/oder Zielort mit Umkreissuche bis 40 km) und Zeitpunkt (beliebig oder konkretes Datum mit Toleranz bis vier Tage möglich). Außerdem können alphabetisch Orte mit bestimmten Anfangsbuchstaben, sowie über entsprechende Kürzel Fahrten bestimmter Länder gesucht werden. Nach der Auswahl wird man zu einer Ergebnisliste weitergeleitet, und bei Interesse können jeweils auf einer **Detailseite** weitergehende Informationen zu einer Fahrt abgerufen werden. Direkt dort sind auch die nötigen (bzw. eingetragenen) Kontaktinformationen des Inserenten zu finden. Diese Infos werden jedoch separat über die „**persönlichen Daten**“ **im Benutzerkonto** verwaltet und automatisch für ein Angebot bzw. Gesuch übernommen.

Eine eigenständige **Buchungsfunktion existiert nicht**, d.h. es gibt auch keine Verwaltung der Platzverteilung. Sicherlich kann dies als Nachteil betrachtet werden, da man sich nicht von vorn herein sicher sein kann, ob die Fahrt überhaupt noch verfügbar ist. Weiterhin ist auch **kein Bewertungssystem** vorzufinden, weshalb es keinen generellen Überblick betreffend Zuverlässigkeit der entsprechenden Nutzer gibt. Jedoch ist für die Zukunft der Einsatz eines solchen Systems geplant, und bis dahin kann die schwarze Liste im Gästebuch genutzt werden, welche in neuer Form als übersichtliche Liste mit Detailansicht zu den einzelnen Einträgen existiert. Da laut Aussage der Betreiber in nächster Zeit mit einem Relaunch der Seite zu rechnen ist, wird sich zeigen, ob eine Bewertungsmöglichkeit angenommen wird, oder ob sich die schwarze Liste bewährt hat.

Grundlegend funktioniert der Dienst ohne Registrierung, d.h. die Suchfunktion kann genutzt werden, um sich über vorhandene Angebote bzw. Gesuche zu informieren (E-Mail Adresse ist jedoch in dem Fall nicht sichtbar). Auch Zusatzfunktionen wie ein Gästebuch sind vorhanden und können zur Ansicht genutzt werden. Weitergehende Funktionalitäten, wie etwa das **Eintragen von Inseraten**, oder alle personenbezogenen Verwaltungsoptionen sind **erst nach der Registrierung** verfügbar. [A]

1.1.2 Drive2Day

- Screenshot:

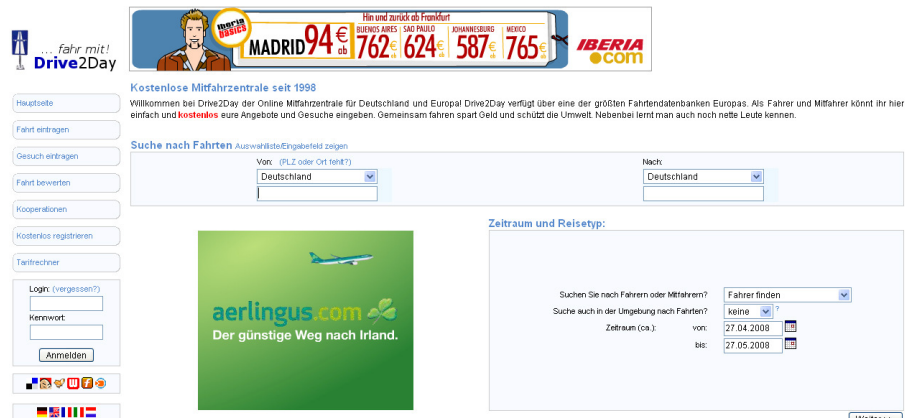


Abbildung 1-2 Screenshot Startseite www.drive2day.de

- Funktionsweise:

Diese Seite bietet schon seit 1998 einen kostenlosen Mitfahrerservice an, der heutzutage unter **www.drive2day.de** zu erreichen ist, und welcher allem Anschein nach gut angenommen wird, da man hier eine große Anzahl verfügbarer Angebote bzw. Gesuche vorfindet. Der strukturelle Aufbau kann auf den ersten Blick aber etwas unübersichtlich wirken. Dies liegt möglicherweise auch daran, dass auf der Startseite in der Bildmitte eine Werbeeinblendung auftaucht, welche die dort befindliche Suchmaske optisch zerteilt. Außerdem ist die Suchfunktion nicht separat in der Menüleiste aufgeführt (sie ist nur auf der Startseite eingebunden), was im ersten Moment etwas verwirrend wirken kann. Ansonsten ist die Seite trotzdem recht nutzerfreundlich und durch die gewählte Farbgebung auch optisch ansprechend.

Die **Eintragung von Inseraten** erfolgt bei Gesuchen in zwei bzw. bei Angeboten in drei Schritten. Zuerst werden Start- und Zielort (Texteingabe oder Liste) mit zugehörigem Land, sowie Art des Inserats (Angebot oder Gesuch) gewählt. Danach erfolgt die Eingabe der konkreten Details. Als Pflichteingabe finden sich hier die Auswahl der zugehörigen Frequenz (einmalig, täglich, wöchentlich, 14-tägig mit den entsprechenden Wochentagen als Einträge in einer einzigen Liste ohne separate Auswahl der Tage), Angabe von Datum und Uhrzeit, sowie die Anzahl der freien bzw. verfügbaren Plätze. Optional kann über dieselbe Seite gleichzeitig noch eine Rückfahrt eingetragen werden, und auch die Angabe von speziellen Fahrtoptionen ist möglich, wie etwa Präferenzen zum Thema Mitnahme von Gepäck, Tieren, Rauchern etc. Bei Gesuchen gibt es auch die Möglichkeit eine Mail-Benachrichtigung einzurichten. Auch bei dieser Seite können wieder zusätzlich Kommentare eingefügt werden, etwa für weitergehende Mitteilungen an eventuelle Interessenten. Im dritten Teil der Eingabe können optional noch bis zu 10 Zwischenhaltepunkte gewählt werden, wobei eine nebenstehende Deutschlandkarte als grobe Orientierungshilfe dient. Die größere Anzahl der Zwischenstationen erhöht beidseitig die Vermittlungschancen, und ist besonders bei längeren Fahrten sinnvoll. Insgesamt kann die kompakte Eingabe der Detaildaten als sehr gelungen betrachtet werden.

Eine **Eingabeprüfung** findet man auch bei diesem Dienst, fehlende Pflichteingaben werden angemerkt und/oder farblich hervorgehoben. Zugriff auf alle persönlich erstellten

Angebote und Gesuche erhält der Nutzer mittels einer praktischen **Übersichtsseite**, durch welche man alle Einträge schnell und einfach administrieren kann. Das heißt, Status ändern (aktiv/inaktiv), Fahrt verlängern, Fahrt bearbeiten oder Fahrt löschen. Hierbei sind ein paar Besonderheiten zu erkennen: Inserate können bei Bedarf inaktiv geschaltet werden, das heißt sie sind für andere Personen nicht mehr auffindbar, müssen aber deswegen nicht direkt gelöscht werden (spätere Änderungen möglich). Weiterhin haben alle Fahrten eine begrenzte Gültigkeit von drei Monaten (betrifft regelmäßige Fahrten); werden diese nicht verlängert, wird der Eintrag zunächst inaktiv gesetzt, und weitere drei Monate später gelöscht. Dies bewahrt den Dienst vor „Inseratsleichen“, das heißt Angeboten oder Gesuchen die nicht mehr gültig sind, aber vom Inserenten nicht selbst gelöscht wurden. Außerdem werden Doppelseinträge automatisch gelöscht; dies passiert wenn ein soeben erfasstes Inserat in Datum, Startort und Zielort einem bereits vorhandenen gleicht (eine Fahrt zu kopieren zwecks neuer Eintragung ist demnach nicht möglich). Zwar ist eine Kopierfunktion äußerst praktisch bei häufigen Eintragungen mit demselben bzw. ähnlichen Inhalt, jedoch ist dies hier nicht unbedingt nötig, da nur wenige Parameter exklusiv über dieses Formular eingetragen werden müssen. Für alle **persönlichen Daten**, wie etwa Kontaktinformationen, Fahrtoptionen und Details zum Fahrzeug gibt es eine **separate Seite** im Bereich der personenbezogenen Einstellungen. Einige dieser Details werden jedoch in die Eingabemaske übernommen und können dort nochmals angepasst werden. Eigenständige Profile tragen zur Übersichtlichkeit bei, und sind deshalb als ein weiterer Beitrag zur Nutzerfreundlichkeit zu betrachten.

Die **Suchmaske** ist zwar durch die eingeblendete Werbung optisch auseinander gezogen, aber die Funktionalität ist trotzdem einfach und effizient. Orte können per Hand eingegeben oder aus einer Liste ausgewählt werden (Eingabe per Postleitzahl auch möglich!). Jedoch werden nur Orte akzeptiert, die sich in der Datenbank befinden; sollte ein gewünschter Ort nicht dabei sein, haben die Nutzer die Chance, per Mail Kontakt mit dem Dienstbetreiber aufzunehmen. Weiterhin können in der Suchmaske die Art der Suche (Fahrer, Mitfahrer oder beides), der Zeitraum (Datum von/bis) und die Umgebungssuche spezifiziert werden. Die Suchergebnisse werden leicht überschaubar aufgelistet, und zu jedem Eintrag kann zusätzlich die entsprechende **Detailseite** aufgerufen werden. Dort sind alle relevanten Daten zum Inserat und Kontaktinformationen zum Nutzer (inklusive Telefonnummer und Kontaktmöglichkeit per Formular - E-Mail Adresse nicht direkt sichtbar), sowie die angegebenen Fahrtoptionen und bisherige Bewertungen dargestellt.

Das **Bewertungssystem** funktioniert nach einem 6-stufigen Schema („super“ bis „mies“), und ist über eine personenbezogene Nutzerkennung geschützt, die man nur durch persönlichen Kontakt mit der betreffenden Person erhält. Außerdem muss man angemeldet sein um Bewertungen vornehmen zu können. Bisherige Bewertungen sind in einer Übersicht im persönlichen Bereich aufgelistet. Jedoch bleiben bei diesem Dienst die Bewertungen anonym. Für den Fall, dass die Kennung an Dritte weitergegeben wurde, könnte sich dieses Merkmal jedoch als äußerst nachteilig erweisen. Sicherheit durch eine eindeutige Nutzer/Nutzer Zuordnung fehlt also. Auffällig ist außerdem die bisher sehr geringfügige Nutzung dieses Merkmals (auch mehrere Jahre nach der Analyse für die Vorarbeit von Sandro Gerstenberger hat sich diese Tatsache nicht geändert). Ein **Buchungssystem existiert** auch bei dieser Seite **nicht**, deshalb gibt es auch keine Regelung der Platzverteilung.

Insgesamt kann dieser Mitfahrdienst jedoch trotz kleinerer Fehler überzeugen (was sich ja auch in der weiterhin hohen Nutzerzahl zeigt), und die unterstützenden Hilfetexte an wichtigen Stellen sind ein nettes kleines Extra! Abschließend sei erwähnt, die grundlegende Funktionalität (Suchen nach bestimmten Mitfahrgelegenheiten) ist auch hier ohne Registrierung zugänglich. [B]

1.1.3 raumobil

- Screenshot:



Abbildung 1-3 Screenshot Startseite www.raumobil.de

- Funktionsweise:

Dieses Dienstportal ist unter **www.raumobil.de** erreichbar. Die Bezeichnung „Portal“ ist in diesem Fall absolut gerechtfertigt, denn hierbei handelt es sich um mehr als einen einfachen Mitfahrdienst. Auf dieser Seite vereinigen sich ein Dienst für die Vermittlung von Mitfahrgelegenheiten (Auto und Bahn), Transportmöglichkeiten, Lagerplätzen und Übernachtungsmöglichkeiten - also ein wahres „Multitalent“ unter den kostenlosen personenbezogenen Transport- und Raumvermittlungen. Es finden sich auch Möglichkeiten zur Vermittlung von Flughafentransfers, Mietwagen, Parkplätzen, Hotels und Bahntickets. Diese sind jedoch größtenteils kommerziell, aber auch eindeutig als solche ausgewiesen (die Dienste an sich, mit allen Funktionen sind jedoch in allen Fällen kostenlos). Es muss aber auch gesagt werden, dass trotz der großen Vielfalt an Funktionalität dieses Portal niemals zu überladen oder kompliziert wirkt. Eine übersichtliche Navigation, klare Symbole und eine angenehme Farbgebung sind nur ein paar der Merkmale, die wesentlich zur Nutzerfreundlichkeit beitragen und damit wahrscheinlich auch mitverantwortlich sind für die wachsende Beliebtheit dieses Portals.

Das bekannte grundlegende Schema – **Inserate eintragen**/über die Suche auf diese Informationen zugreifen – ist auch hier wieder vorzufinden. Alle Einträge können ohne Registrierung eingesehen werden, aber eigene Inserate sind auch hier wieder **nur durch vorherige Registrierung möglich**. Um ein solches Inserat (mit Eingabepprüfung) zu erstellen, müssen drei Schritte befolgt werden. Zuerst werden Art (Fahrer/Mitfahrer), Häufigkeit (einmalig/regelmäßig), Startort/Zielort (mit Länderwahl aus Liste und Umkreissuche bis 20 km; Ortseingabe auch mit Postleitzahl, per Hand mit Eingabehilfe) mit bis zu 5 Zwischenstopps und der Grund des Inserats (Person oder Dinge) abgefragt. Sehr positiv ist auf jeden Fall die große Anzahl der durch die Datenbank vorgegebenen Orte. Selbst kleinere Ziele sind möglich, was einen wesentlichen Vorteil gegenüber anderen Diensten mit Vorgabe darstellt. Im nächsten Schritt erfolgt die Angabe der Personenzahl (möglich bzw. gesucht), Präferenzen (optional) betreffend Raucher und Geschlecht der Mitreisenden, des Datums und der Uhrzeit bei einmaligen Inseraten, der Uhrzeit und konkreten Frequenz bei regelmäßigen Inseraten (täglich, werktäglich, wöchentlich mit

Angabe der gültigen Wochentage), sowie des genutzten Verkehrsmittels (PKW, LKW, Kleintransporter, Bus, Bahn, Schiff, Flugzeug, sonstiges). Zum Schluss kann noch der Fahrtpreis festgelegt werden (oder „nach Absprache“), außerdem hat man die Möglichkeit ein Bild (bspw. vom Fahrzeug) einzufügen, eine Rückfahrt anzubieten oder zusätzliche Bemerkungen zu verfassen. Obwohl die Eingabe insgesamt weniger kompakt erscheint, ist diese durch ihre logische Gliederung trotzdem effektiv. Abgeschlossene Inserate können in einer **Übersicht** im persönlichen Bereich administriert werden (aktiv/inaktiv schalten, Bearbeiten, Kopieren oder Löschen), wobei auch hier wieder zwischen Personen und Räumen unterschieden wird.

Die **Suchfunktion** ist bei dieser Seite zweigeteilt. Auf der Startseite findet man eine Art „Schnellsuche“ mit Auswahl von Art (Person/Raum) sowie Start- und/oder Zielort (leere Eingabe bedeutet „egal“). Weiter unten auf dieser Seite ist auch eine alphabetisch geordnete Suche nach Orten mit bestimmten Anfangsbuchstaben möglich. Über den entsprechenden Menüpunkt kann bei Bedarf die Detailsuche aufgerufen werden, wobei hier jeweils eine separate Suche für Personen und Räume existiert. Folgende Parameter stehen betreffend Mitfahrgelegenheiten zur Wahl: Art (Fahrer/Mitfahrer), Start- und Zielort (Eingabe wie beim Inserieren), sowie Datum (konkrete Angabe oder beliebig möglich). Auch hierbei zeigt sich die Abgrenzung der einzelnen angebotenen Dienstsegmente, was für ein Portal mit solch einem Umfang auch nicht zu vermeiden ist, um eine unkomplizierte Nutzung zu garantieren. Die Suchergebnisse werden übersichtlich aufgelistet und alle wesentlichen Merkmale sind über verständliche Symbole zu unterscheiden. **Detailinformationen** sind jeweils auf einer einzelnen Seite aufgeführt. Über diese gelangt man auch zu einem **Kontaktformular** und der **persönlichen Profilseite**. Alle angegebenen, nicht das konkrete Inserat betreffenden, das heißt personenbezogenen Daten sind hier getrennt aufgeführt. Dazu zählen Dinge angefangen von Telefonnummern, Angaben zum Fahrzeug, anwendbaren Sprachen, Raucherstatus, etc. bis hin zu Hobby und Beruf. Außerdem kann ein Foto hochgeladen werden, was für die leichtere Identifizierung durchaus sinnvoll ist. Die Verwaltung der Profildaten erfolgt wiederum im persönlichen Nutzerbereich.

Ein **Bewertungssystem** ist bisher **nicht implementiert**, soll aber nach eigener Angabe demnächst eingeführt werden. Leider gibt es momentan aber auch keine wirkliche Ausweichlösung, wie etwa ein Gästebuch, um negativ aufgefallene Fahrer bzw. Mitfahrer für unwissende Nutzer erkennbar zu machen.

Bei diesem Mitfahrdienst ist auch **kein Buchungssystem** vorhanden. Bei Interesse muss man mit dem Inserenten Kontakt aufnehmen, um sich für eine bestimmte Fahrt zu melden. Eine Platzverwaltung gibt es demnach aber auch nicht, was sich bei beliebten bzw. oft frequentierten Strecken als problematisch erweisen kann.

Insgesamt ist dieses Dienstportal eindeutig als das Vielfältigste seiner Art zu bezeichnen, also eine prima Anlaufstelle für verschiedenste Dienstleistungsbedürfnisse von Studenten. Trotz der Vielfalt stellt sich die Nutzung als unkompliziert und übersichtlich dar, und durch diese gelungene Kombination wird diese Seite seinen Nutzerstamm sicherlich noch ausbauen können, wobei sich die Funktionalität auch ständig weiterentwickelt und verbessert, im Gegensatz zu manch anderem Mitfahrservice der sich seit Jahren etwa in der selben Entwicklungsstufe befindet. [C]

1.1.4 Mitfahrzentrale

- Screenshot:

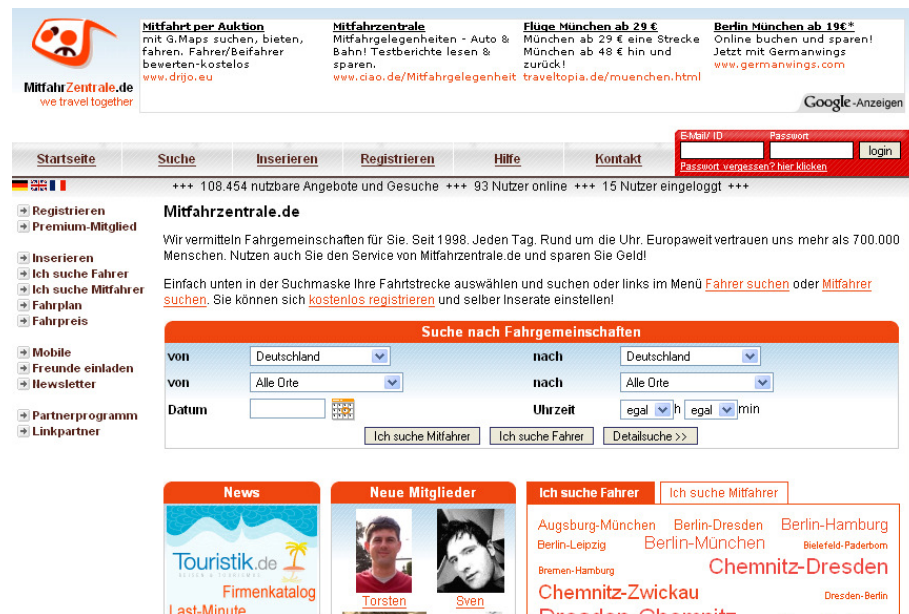


Abbildung 1-4 Screenshot Startseite www.mitfahrzentrale.de

- Funktionsweise:

Hierbei handelt es sich um einen kostenlos angebotenen, aber kommerziell ausgerichteten Dienst, welcher unter **www.mitfahrzentrale.de** zu erreichen ist, 1998 gegründet wurde und europaweit genutzt wird. Trotz der eingeschränkten Übersichtlichkeit aufgrund vorhandener Werbeeinblendungen kann sich der Nutzer noch gut orientieren. Direkt erkennbar in der Bildmitte ist die **einfache Suchmaske** platziert, was einem neuen Nutzer, oder jemandem der sich nur schnell informieren will, sehr entgegenkommt. Gesucht werden kann hier nach Start- und/oder Zielort, auch Postleitzahlen können eingegeben werden (inklusive entsprechender Auswahl des betreffenden Landes), wobei auch die Option „alle Orte“ existiert. Diese Eigenschaft ist sinnvoll, um sich beispielsweise alle Fahrten eines bestimmten Start- oder Zielortes anzeigen zu lassen (etwa: welche Fahrten existieren für meinen Heimatort?). Ergänzend können auch konkrete Angaben zu Datum und Uhrzeit gemacht werden. Die Auswahl der Art erfolgt über die eingblendeten Buttons. Die **Detailsuche** ist bei Bedarf über den entsprechenden Menüpunkt aufzurufen. Dort können interessierte Nutzer ihre Anfrage mittels spezieller Parameter konkretisieren. Zusätzlich mögliche Angaben beinhalten eine Toleranzangabe für Datum und Uhrzeit, sowie die Option, nur Inserate von Nichtraucher zu zeigen. Auch im Umkreis bis 50 km kann gesucht werden, wobei auch ohne diese Angabe eine Umkreissuche durchgeführt wird, dessen Ergebnisvorschläge am Seitenende aufgeführt sind. Es gibt auch die Möglichkeit einer alphabetischen Ortsauflistung, oder der Suche nach Fahrten bestimmter Länder (über alphabetisch sortierte Länderkürzel). Über spezielle Menüpunkte auf der linken Seite können auch direkt alle eingetragenen, gültigen Angebote oder Gesuche aufgezeigt werden. Die Suchergebnisse werden erst nach Datum/Frequenz, dann nach Strecke und Uhrzeit sortiert aufgelistet (bei alphabetischer Suche nach Ort und Land sortiert). Trägt man regelmäßige Inserate ein, werden diese für jeden Gültigkeitstag einzeln in der Suche aufgeführt. Da hier aber keine Platzverwaltung existiert, sollte das kein größeres Problem darstellen. Einerseits vereinfacht dies die

Vermittlung für Personen, die nur an bestimmten Tagen fahren möchten. Andererseits jedoch könnten die einzeln aufgeführten Tage fälschlicherweise als separate Inserate wahrgenommen werden, was zu Verwirrung führen könnte für den Fall einer schon ausgebuchten Fahrt. In dem Fall gäbe es eine große Menge an Einträgen, die alle nicht mehr in Frage kommen, und weil der Inserent erst in der Detailansicht preisgegeben wird, ist die Navigation durch alle Inserate ziemlich umständlich. Ein kompakter Eintrag, oder zumindest die direkte Anzeige des Inserenten in der Ergebnisliste wären wahrscheinlich die besseren Alternativen.

Die konkreten Einzelheiten eines Eintrags sind entsprechend in einer **Detailansicht**, das heißt auf einer Einzelseite aufgeführt. Dort sind außer den relevanten fahrtbezogenen Einzelheiten auch Informationen über den Inserenten aufgeführt. Dinge wie Telefonnummern, E-Mail Adresse (nicht direkt sichtbar, Kontakt nur über ein Formular möglich), oder auch persönliche Eigenschaften zum Thema Raucher, nutzbare Sprachen und ein Foto können eingesehen werden. Diese werden über ein separates **Nutzerprofil** (im Nutzerkonto) eingetragen, und teilweise für die Inseratsdarstellung übernommen. Eine Besonderheit bei diesem Dienst ist sicherlich die sogenannte Profilkarte, auf der optional noch genauere Angaben zur Person gemacht werden können (bspw. Beruf, Sternzeichen, oder eine allgemeine Beschreibung). Außerdem existiert ein **separates Profil** für umfangreiche Angaben **zum eigenen Fahrzeug**. Folgende Parameter sind dabei als Pflichteingabe zu betrachten: Kennzeichen, Marke, Modell und Farbe. Zusätzlich können noch weitere optionale Angaben zur Ausstattung gemacht oder sogar ein Bild hochgeladen werden. Interessierte Nutzer können, ein konkretes Inserat betreffend, über ein **Kontaktformular** oder eine kostenpflichtige SMS den Inserenten erreichen. Die angegebenen Telefonnummern müssen bei diesem Dienst aber kostenpflichtig angefordert werden. Gegen eine monatliche Gebühr kann auch die „Premium-Mitgliedschaft“ erworben werden. In diesem Fall sind unter anderem alle Telefonnummern (auch die eigenen) immer frei sichtbar, natürlich nur so lange gezahlt wird.

Die **Eingabe von Angeboten bzw. Gesuchen** kann auf zwei verschiedene Arten ausgeführt werden. In der älteren Variante geschieht dies in vier Schritten mittels aufeinanderfolgenden Formularseiten. Die neuere Variante ist kompakter und wirkt wesentlich übersichtlicher. Hierbei gibt es nur eine Formularseite, welche aber in drei logische Abschnitte unterteilt ist. Pflichteingaben beschränken sich auf das nötigste: Strecke (und Land; jedoch ohne Zwischenhaltepunkte), Art (Angebot/Gesuch), Frequenz (einmalig, oder regelmäßig mit Auswahl der betreffenden Wochentage), Datum, Uhrzeit, Fahrpreis (oder Verhandlungssache) und Plätze. Optional können noch zusätzliche Bemerkungen verfasst, sowie Angaben zu persönlichen Präferenzen getroffen werden. Die abgeschlossenen Eintragungen können in einer **Übersicht im Nutzerkonto** verwaltet werden (aktiv/inaktiv schalten, bearbeiten, löschen; kopieren ist nur direkt nach der Eintragung möglich) und in diesem Fall werden einmalige und regelmäßige Fahrten getrennt aufgelistet. Im Nutzerkonto ist es unter anderem auch möglich Fahrterinnerungen anzulegen, Treffenbenachrichtigungen abzuschalten und allgemeine Nachrichten zu verwalten.

Dieser Mitfahrdienst bietet seinen Nutzern außerdem die Möglichkeit der gegenseitigen **Bewertung**. Zu Grunde liegt dabei ein dreistufiges System (positiv, neutral, negativ), welches zwar keine Beurteilung mittels differenzierter Kriterien ermöglicht, aber ein zusätzliches Kommentarfeld anbietet. Einen konkreten Missbrauchsschutz in dem Sinne gibt es nicht, jedoch werden die Bewertungen nicht anonym vorgenommen, und eine vorherige Registrierung ist Pflicht (danach muss die betreffende Fahrt vorgemerkt wer-

den, um später bewertet werden zu können). Bisher wurde die Bewertungsfunktion jedoch sehr selten genutzt.

Weiterhin ist dieser Dienst der einzige von mir untersuchte, welcher eine Art **Buchung** anbietet. Diese betrifft aber nur Fahrten der Art „Gesuch“, und ist ein weiterer kommerzieller Aspekt dieser Seite. Dem Inserenten wird eine SMS mit einem Code geschickt, ausgelöst von einem buchungswilligen Nutzer. Wird dieser Code in den folgenden zwei Stunden zurückgeschickt, gilt die Buchung als bestätigt. Daraufhin wird das Inserat aus der Datenbank gelöscht. Jedoch könnte sich dies als nachteilig erweisen falls die Buchung zurückgezogen wird, denn das zugehörige Inserat müsste dann neu erstellt werden und potentielle Interessenten hätten in der Zwischenzeit nichts von dem Gesuch gewusst.

Insgesamt handelt es bei dem vorliegenden Dienst um eine gelungene Umsetzung eines Mitfahrservice, der in den grundlegendsten Funktionen kostenfrei nutzbar ist (was wahrscheinlich unumgänglich ist, um eine breite Masse an Nutzern anzulocken), aber auch Zusatzleistungen gegen Gebühr anbietet. Die große Zahl an eingetragenen Fahrten macht diesen Dienst zu einer bevorzugten Anlaufstelle für die Vermittlung von Mitfahrgelegenheiten. Hervorzuheben ist das umfangreiche Hilfsmenü, mit Schritt für Schritt Anleitungen zu den einzelnen Funktionen. [D]

1.1.5 MFZ

- Screenshot:



Abbildung 1-5 Screenshot Startseite www.mfz.de

- Funktionsweise:

Dieser kommerziell orientierte Mitfahrdienst ist unter **www.mfz.de** erreichbar. Obwohl die optische Aufmachung das erst einmal nicht sofort vermuten lässt, ist das zu Grunde liegende System dasselbe wie vom Dienst „Mitfahrzentrale“ (mit ein paar Variationen). Insgesamt ist dieser Dienst etwas simpler, greift aber auf denselben Datenbestand zu. Falls man beispielsweise bei www.mitfahrzentrale.de ein Inserat erstellt hat, taucht dieses auch hier im Fahrtmanager auf, oder ist über die Suche aufzufinden (funktioniert auch anders herum). Bei der Umsetzung gibt es ein paar Gemeinsamkeiten, wie beispielsweise das komplette Nutzerkonto (inklusive Fahrtmanager, Profildaten und Nachrichtenverwaltung). Auch die einfache Suche auf der Startseite ist identisch, was ebenso auf dessen Ergebnisliste zutrifft. Die aufgeführten Einzelheiten der Fahrtdetails sind ebenfalls größtenteils identisch, jedoch ist die grafische Darstellung an diesen Dienst angepasst worden.

Neben den wesentlichen Gemeinsamkeiten gibt es aber auch Unterschiede zwischen den beiden Diensten, wobei man diese hier eher als Vereinfachungen bezeichnen kann.

Es existiert **keine Detailsuche**, auch ein **Bewertungssystem** ist in diesem „Interface“ **nicht implementiert**. Diese Funktion ist geplant, aber bisher nicht realisiert worden. Deshalb hat die Bewertungsübersicht im Nutzerkonto im Bezug auf diesen Dienst auch noch keinen konkreten Nutzen. Weiterhin ist für diesen Mitfahrdienst die „quasi **Buchungsfunktion**“ für Gesuche **nicht vorhanden**. Das Hilfemenü ist wesentlich vereinfacht worden, und bei der **Eingabemaske für Inserate** ist nur die ältere Variante verfügbar. Das heißt, die Eingabe erfolgt zuerst einmal über vier aufeinanderfolgende Formularseiten. Als erstes muss man sich für eine Art des zu erstellenden Inserats entscheiden – Angebot, Gesuch oder beides („Ich kann selbst fahren, würde aber auch bei jemand

anderem mitfahren“). Auf der nächsten Seite werden Start- und Zielort (mit zugehörigem Land) aus einer Liste ausgewählt, die allgemeine Frequenz angegeben (einmalig oder regelmäßig) und die Rückfahrtsoption gesetzt. Im selben Schritt können auch noch optional die persönlichen Präferenzen spezifiziert werden (was bei der kompakten Variante zum Schluss stand). Die genauen Details zur Fahrt werden in Schritt drei konkretisiert. Bei regelmäßigen Inseraten muss man die gültigen Wochentage festlegen (wobei die Uhrzeit für jeden Tag separat gewählt werden kann), sowie den Wochenrhythmus (wöchentlich, 14-tägig, 4-wöchentlich) und die Gültigkeitsdauer (1-4 Monate) wählen. Bei einmaligen Inseraten muss man sich nur noch für die Uhrzeit entscheiden (Option „egal“ ist möglich). Weiterhin können hier das Datum der Fahrt, der Fahrpreis (nur bei Angeboten; kein „egal“ möglich) sowie die verfügbaren oder benötigten Plätze angegeben und zusätzliche Bemerkungen verfasst werden. In Schritt vier ist die Eingabe eigentlich bereits abgeschlossen, es wird nur noch auf die nächsten möglichen Optionen im Bezug auf das erstellte Inserat hingewiesen (bspw. die **Administration im Fahrtmanager**). Bei dieser verteilten Eingabe geht jedoch die Übersicht relativ schnell verloren, weswegen ein kompaktes und logisch geordnetes Formular (wie etwa bei „Mitfahrzentrale“ vorhanden) einen wesentlichen Beitrag zur Nutzerfreundlichkeit leisten würde.

Obwohl diese Realisierung eines Mitfahrservice grob gesagt als eine vereinfachte Variante des Dienstes „Mitfahrzentrale“ bezeichnet werden kann und hierbei sogar die Werbung optisch weniger auffallend wirkt, ist damit nicht gesagt, dass auch die Nutzung einfacher oder angenehmer ist. Im Gegenteil, der Dienst auf www.mitfahrzentrale.de wirkt insgesamt doch etwas bequemer und professioneller. [E]

1.1.6 Liftpool

- Screenshot:

The screenshot shows the Liftpool website interface. At the top, it says 'Die kostenlose Mitfahrzentrale für Deutschland und Europa. Mitfahrbörse - Mitfahrgelegenheit'. The navigation bar includes 'Eingeloggt als: mooby85', 'Abmelden', and the date 'Samstag, 26.04.2008'. The left sidebar lists various menu items like 'Home', 'Registrieren', 'Profisuche', etc. The main content area is titled 'Profisuche' and features a search form for 'MITFAHRER'. The form has two sections: 'Deutschland - Nur große Städte' and 'Deutschland - Alle Orte'. Each section has dropdown menus for 'von' and 'nach' (both set to 'Alle großen Städte' or 'Alle Orte'), radio buttons for 'Fahrangebote' and 'Fahrgesuche', and date/tolerance selection. The right sidebar contains a 'liftpool network' logo, language options, 'Google-Anzeigen', and links to 'Fracht Transporte' and 'Reisen nach Lettland'.

Abbildung 1-6 Screenshot Profisuche www.liftpool.de

- Funktionsweise:

Unter www.liftpool.de erreicht man diesen europaweit nutzbaren Mitfahrdienst. Der erste Eindruck dieser Seite ist äußerst positiv, denn die verwendete Farbgebung wirkt sehr freundlich und ist angenehm für die Augen. Eine Vielzahl verfügbarer Sprachen ist ein weiterer Pluspunkt dieses Dienstes. Jedoch fallen dem Nutzer auch sehr schnell einige gravierende Fehler auf dieser Seite auf. Was sehr schade ist, da dieses Nutzerinterface an sich durch seine angenehme und übersichtliche Bedienung durchaus Potential besitzt, ein beliebter Anlaufpunkt unter den verfügbaren Mitfahrdiensten zu sein. Aber schon bei der Betrachtung aller eingetragenen Inserate (über das Menü können alle Angebote bzw. Gesuche angezeigt werden) wird klar, dass es hier Probleme gibt, denn es sind generell nur sehr wenige Eintragungen vorhanden. Aber es lohnt sich natürlich trotzdem, das grundlegende Schema dieses Dienstes zu analysieren.

Wie bei allen anderen von mir analysierten Diensten, kann auch dieser hier ohne Registrierung genutzt werden, um eingetragene Inserate zu recherchieren. Möchte ein Nutzer selbst ein Angebot oder Gesuch aufgeben, muss er sich registrieren. Danach kann über das Nutzerkonto auch ein persönliches Profil eingetragen werden, welches Platz für eine Vielzahl von Kontaktdaten zum Nutzer enthält. Pflichtdaten sind dabei Name, Adresse, E-Mail und Telefonnummer. Optional können weitere Angaben gemacht werden, wie etwa die Mobilfunknummer, die Faxnummer, ICQ, Homepage und zusätzliche Kommentare. Eine Besonderheit ist hierbei die Möglichkeit der (teilweisen) Auswahl, welche Angaben davon offen für andere Nutzer sichtbar sind. Diese Details werden dann bei der Eintragung eines Inserats übernommen.

Diese **Eintragung** erfolgt in drei Schritten mittels aufeinanderfolgenden Formularseiten. Zuerst werden die entsprechenden Länder von Start- und Zielort gewählt. Schritt zwei beinhaltet die Auswahl eben dieser Orte und die grundlegende Häufigkeit der Fahrt (Einzelfahrt oder Mehrfachfahrt). Hierbei können wieder nur Orte aus einer Liste ausgewählt werden (also keine eigene Angabe möglich) und die Auswahl ist zweigeteilt. Einerseits gibt es die Liste für größere Orte, andererseits gibt es aber noch eine zweite Liste für „alle Orte“ (aus einer Stichprobe geht hervor, dass tatsächlich viele kleine Orte und Dörfer vorzufinden sind). Beide Listen sind alphabetisch sortiert, wobei zuerst der betreffende Anfangsbuchstabe gewählt werden muss. Im letzten Schritt erfolgt dann die Angabe der Fahrtdetails. Neben den standardmäßigen Angaben wie Datum, Uhrzeit (Start- und Ankunftszeit), freien oder benötigten Plätzen, zusätzlichen Bemerkungen sowie betreffenden Fahrttagen und Gültigkeitsdauer bei regelmäßigen Inseraten, können auch noch konkrete **Angaben zum Fahrzeug** gemacht werden. Hierfür gibt es jedoch **kein separates Profil**, was umständlich ist, falls häufiger Eintragungen vorgenommen werden sollten. Obwohl die Inseratseingabe nicht kompakt auf einer Seite stattfindet, wird durch die logische Abfolge der Angaben trotzdem die Übersichtlichkeit bewahrt.

Alle eingetragenen Inserate können in einer **Übersicht im Nutzerkonto** verwaltet werden. Zu bemerken ist bei diesem Dienst die grundsätzliche Trennung von Angeboten und Gesuchen. Folgende Verwaltungsfunktionen können genutzt werden: Details betrachten, Editieren, Duplizieren und Löschen. Ein wesentlicher Nachteil bei der Handhabung der Eintragungen ist jedoch die Einzelaufstellung aller Gültigkeitstage. Dies gilt hier nicht nur für die Suche, sondern auch für die Verwaltung. Hat man beispielsweise ein regelmäßiges Inserat, welches montags bis freitags angeboten wird, für mehrere Wochen Gültigkeit eingetragen, und möchte dieses nun nicht mehr anbieten, müsste man jeden Gültigkeitstag für jedes eingetragene Datum einzeln löschen.

Die **Suchfunktion**, hier „Profisuche“ genannt, ist sehr ausführlich, aber dafür in mehrere Abschnitte unterteilt. Ähnlich wie bei der Eintragung, kann auch hier nach größeren oder allen Orten gesucht werden - inklusive Spezifizierung der Art (Fahrangebote/Fahrgesuche), des Datums und zugehöriger Toleranz (auch „beliebig“ möglich), sowie einer Umkreissuche. Es kann auch nach einzelnen Ländern, alphabetisch nach Abfahrtsländern und alphabetisch nach Abfahrts- oder Zielort gesucht werden. Alle Ergebnisse werden übersichtlich aufgelistet und zu jedem Eintrag können die angegebenen **Details separat abgerufen** werden.

Insgesamt besitzt dieser Mitfahrdienst definitiv Potential, jedoch gibt es Mängel in der Umsetzung. Beispielsweise kann bei der Eintragung eines einmaligen Inserats nicht das aktuelle Jahr ausgewählt werden, weswegen ein solcher Eintrag sofort verworfen wird. Deshalb hat dieser Dienst fast keine Nutzer mehr, und ist demnach auch nicht weiterzuempfehlen. [F]

1.1.7 Detailauflistung

Im Folgenden werden die sechs analysierten Dienste im Detail mittels ausgewählter Kriterien nach verschiedenen Kategorien geordnet verglichen:

Legende:

- X - trifft zu
- O - trifft bedingt zu (Bemerkung beachten)
- / - trifft nicht zu

▪ Allgemeine Merkmale:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
Registrierung erforderlich	O (Anschauen nein; Eintragen, etc. ja)	O (Anschauen nein; Eintragen, Bewerten etc. ja)	O (Anschauen nein; Eintragen, etc. ja)	O (Anschauen nein; Eintragen, Bewerten etc. ja)	O (Anschauen nein; Eintragen, etc. ja)	O (Anschauen nein; Eintragen, etc. ja)
gebührenpflichtig	/	/	/	O (Grundfunktionen frei; erweiterte Funktionalität kostenpflichtig)	O (Grundfunktionen frei; erweiterte Funktionalität kostenpflichtig)	/
mehrsprachig	/ (englisch in Arbeit)	X (eng, fr, ita, nl)	/	X (eng. über Partnerseite)	/	X (eng, fr, ita, ung, pol, spa, Esperanto)
WAP fähig	X	/	/ (in Planung)	X (+ i-mode)	/	/
SMS fähig	/	/	/	X (nur als zahlendes Premium-Mitglied)	/	/
Fahrpreisrechner	X	X	X	X	X	/
Pers. Schwarzes Brett/Startseite	/	/	/	/	/	/
„aktuelle“ Inserate (Startseite)	/	/	X (neue Fahrten)	O (tag-cloud mit konkreten Strecken)	/	/
Routenplaner	/	/	O (Google Maps Link im Inserat)	O (Google Maps Link im Inserat)	/	/ (nur Verweise auf Fremdanbieter)
Nachrichten schicken	/	X	X	X	X	/
Forum/Gästebuch	X	/	O („User Tipps“- schwarzes Brett)	/	/	O (derzeit gesperrt)
Missbrauch melden	X (+ „Schwarze Liste“)	/ (Bewertung nutzen !)	X	O (Bewertung nutzen; E-Mail an Support)	/ (Bewertung nutzen !)	/
Weitere (ähnliche Dienste)	/ (Bahn Mitfahrdienst in Arbeit)	/	X (Bahn, Flughafentransfer, Transport, Raum, Mietwagen, Parkplatz; Flüge) – teils kommerziell	/	/	/

Tabelle 1-1 Vergleich vorhandener Mitfahrdienste - allgemeine Merkmale

▪ Suche:

	Mitfahrgelegenheit	Drive2Day	raumobil	mitfahrzentrale	MFZ	Liftpool
Suchmaske (einfach)	X (Startseite)	/	X (Startseite)	X (Startseite)	X (Startseite)	/
Suchmaske (Detail)	X	X (Startseite)	X	X	/	X
Unterscheidung Angebote/Gesuche	X	X	X	X	X	X
Regelmäßige Angebote/Gesuche	/ (im Inserat erkennbar)	/ (im Inserat erkennbar)	/ (im Inserat erkennbar)	/ (im Inserat erkennbar)	/ (im Inserat erkennbar)	/
Alphabetische Suche	X	/	/	X	/	X
Orte vorgegeben	X	X (+ Feld mit Eingabehilfe)	X	X	X	X
(Ergänzung mögl.)	X (bei Betreiber melden)	X (bei Betreiber melden)	/	/	/	/ (große DB !)
Eigene Ortsangabe mögl.	/	X (mit Hilfe)	X (Detailsuche mit Hilfe)	X (findet aber nur übereinstimmende Orte)	/	/
Alle Orte	X (nur Start- oder Zielort)	X (Auswahl-liste /leeres Feld)	O (leeres Feld)	X	X	X
Zwischenhalte berücksichtigt	X	X	X	/ (Google Maps – was vereinbaren?)	X	/
Umkreissuche	X (10,20,30,40 km)	X (10,20,40 km)	/	X (bis 50; 5er Schritte)	O (automatisch)	X (20,50 km)
Postleitzahlen	/	X	X	O (wenn in Datenbank)	/	/
Kalenderoption	X (+ drop-down)	X (+ Selbst-eingabe)	X (+ Selbsteingabe)	X (+ Selbsteingabe)	X (+ Selbst-eingabe)	/ (nur drop-down)
Datum egal	X	/ (nur von / bis)	X	X	X	X
Toleranz (Datum)	X	/	/	X	/	X
Uhrzeit enthalten	/ (nur Datum)	/ (nur Datum)	/ (im Inserat)	X	X	/ (im Inserat)
Such-Hitliste	/	/	/	/	/	/

Tabelle 1-2 Vergleich vorhandener Mitfahrdienste – Suche

▪ Fahrt(en) eintragen:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
Registrierung erforderlich (Eintragung)	X	X	X	X	X	X
Unterscheidung Angebote/Gesuche	X	X	X	X	X	X
Kalenderoption	X (+ drop-down)	X (+ Selbst-eingabe)	X (+ Selbsteingabe)	X (+ Selbsteingabe)	X (+ Selbst-eingabe)	/ (nur drop-down)
Regelmäßige Angebote/Gesuche	X	X	X	X	X	X
(einzelne Wochentage)	X	X	X	X	X	X
Zwischenstopps	X (3)	X (9)	X (5)	/	/	/
Plätze (benötigt/frei)	X	X	X	X	X	X
Preisangabe	/	X	X	X	X	/
Preisangabe pro Person/Auto	<input type="radio"/> (keine explizite Angabe im Inserat; Empfehlung pro Person)	X (pro Person)	X (pro Person)	X (pro Person)	X (pro Person)	/
Uhrzeit	X	X	X	X	X	X
Treffpunkt	/	/	/	/	/	/
Bemerkungen	X	X	X	X	X	X
Zusatzinformationen (allg.)	X	X	X	X	X	X
Kontaktinfos	X	X	X	X	/ (nur Formular)	X
Änderungen mögl.	X	X	X	X	X	X
Hinweis – zuletzt aktualisiert	X	/	X	X	X	/ (nur Eingabezeitpunkt)
Details drucken	X	/	/	X	X	X
Rückfahrten	X	X	X	X	X	/
Urlaubsfahrten	<input type="radio"/> (normales Inserat nutzen)	<input type="radio"/> (normales Inserat nutzen)	<input type="radio"/> (normales Inserat nutzen)	<input type="radio"/> (normales Inserat nutzen)	<input type="radio"/> (normales Inserat nutzen)	<input type="radio"/> (normales Inserat nutzen)

Tabelle 1-3 Vergleich vorhandener Mitfahrdienste - Fahrt(en) eintragen

▪ Buchungssystem:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
vorhanden	nicht integriert	nicht integriert	nicht integriert	O (Gesuche)	nicht integriert	nicht integriert
Bemerkungen				/		
Zusatzinfos				/		
Benachrichtigungs-funktion				X (SMS)		
Absage mögl.				O (Bestätigungscode nicht zurückschicken)		
Platzverwaltung				/		
Rückwirkung auf Angebo- te/Gesuche				X (Gesuch bei Bestätigung gelöscht)		

Tabelle 1-4 Vergleich vorhandener Mitfahrdienste – Buchungssystem

▪ Bewertungssystem:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
System	nicht integriert → demnächst → (schwarze Liste im Forum)	1-6 (super bis mies)	nicht integriert → demnächst	3 stufig (positiv; neutral; negativ)	nicht integriert → dem- nächst	nicht integriert
Unterscheidung Fahrer/Mitfahrer		X		X		
Verschiedene Kriterien		X		/		
Kommentar				X		
Zeitliche Be- grenzung		/		/		
Missbrauchs- schutz		X (Bewertung anonym, aber nur möglich, wenn ange- meldet; beteiligte Personen müssen Nutzerken- nung überge- ben)		O (erst Fahrt merken, dann nach der Fahrt bewerten - nur wenn registriert)		
Unterscheidung bei regelmässi- gen Fahrten		X (eine Person nur einmal die Woche)		/ (da jede Einzelfahrt separat aufgeführt)		

Tabelle 1-5 Vergleich vorhandener Mitfahrdienste – Bewertungssystem

▪ Nutzerprofil:

	Mitfahrgelegenheit	Drive2Day	raumobil	mitfahrzentrale	MFZ	Liftpool
Festes Nutzerprofil	<input type="radio"/> („nur“ pers. Daten)	X	X	X	X	X
Kontaktinfos	X	X	X	X	X	X
Foto	/	X	X	X	X	/
Zusatzinfos (Raucher etc.)	<input type="radio"/> (über Inserat)	X (+ pers. Voreinstellungen)	X	X	X (teilweise über Inserat)	/
Kfz-Infos	<input type="radio"/> (über Inserat)	/	X	<input type="radio"/> (separates Profil)	/	<input type="radio"/> (über Inserat)
Ohne Anmeldung einsehbar	<input type="radio"/> (über Inserat; E-Mail nur wenn eingeloggt)	<input type="radio"/> (über Inserat)	X	X (Tel.-Nr. gebührenpflichtig)	X (Tel.-Nr. gebührenpflichtig)	<input type="radio"/> (über Inserat)

Tabelle 1-6 Vergleich vorhandener Mitfahrdienste – Nutzerprofil

▪ Fahrzeugprofil:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
möglich	<input type="radio"/> (über Inserat)	nicht integriert	<input type="radio"/> (über pers. Profil)	X	nicht integriert	<input type="radio"/> (über Inserat)
freiwillig	X		X	X		X
ohne Anmeldung einsehbar	X (über Inserat)		X	X		X (über Inserat)
Spezielle Merkmale	X (Kennzeichen)		X (Fahrzeug, Kennzeichen)	X (Kennzeichen, Marke, Modell, Baujahr, Farbe, Foto)		X (Kennzeichen, Marke, Typ, Farbe, Nichtraucherauto)
allg. Beschreibung	/		/	/ (nur spezielle Merkmale + Foto)		/ (nur spezielle Merkmale)
Foto	/		X	X		/

Tabelle 1-7 Vergleich vorhandener Mitfahrdienste – Fahrzeugprofil

▪ Verwaltung:

	Mitfahrgelegenheit	Drive2Day	raumobil	Mitfahrzentrale	MFZ	Liftpool
Angebote/Gesuche Übersicht	X	X	X	X	X	X
Buchungsübersicht	/	/	/	O (nur gemerkte Fahrten)	O (nur gemerkte Fahrten)	/
Bewertungsübersicht	/	X	/	X	/	/
Änderungen mögl.	X	X	X	X	X	X
Platzverwaltung	/	/	/	/	/	/
Reminder Service	X (E-Mail)	X (E-Mail; wenn passendes Gesuch eingetragen)	/	X (Trefferbenachrichtigung; Fahrterinnerungen)	X (Trefferbenachrichtigung; Fahrterinnerungen)	/
Terminkalender	/	/	/	/	/	/

Tabelle 1-8 Vergleich vorhandener Mitfahrdienste - Verwaltung

1.2 Webbasierte Umfrage

Im Zuge der Vorarbeit von Sandro Gerstenberger wurde unter anderem auch eine Umfrage² zu spezifischen Charakteristiken eines möglichen Mitfahrdienstes durchgeführt. Die Studenten der Hochschule Mittweida hatten dabei die Möglichkeit größtenteils web-basiert und teilweise mündlich an dieser Befragung teilzunehmen, wobei sich insgesamt 171 Studenten mit vollständigen Angaben beteiligten. Die Kernaussagen der ermittelten Ergebnisse dieser Umfrage sollen nun herangezogen werden, um als zusätzliches Vergleichskriterium einen weiteren Beitrag zur Ermittlung von allgemeinen Anforderungsmerkmalen eines Mitfahrdienstes zu ergeben.

Die Fragestellungen wurden in drei logische Bereiche unterteilt:

- Allgemeine Fragen zum Dienst
- Fragen aus der Sicht eines Anbieters
- Fragen aus der Sicht eines Mitfahrers

Im ersten Teil (welcher **allgemeine Aspekte** abdeckte) ergab sich, dass fast alle Teilnehmer grundsätzlich einen solchen Dienst nutzen würden (98,2 %), sowohl um Angebote für potentielle Mitfahrer (19,3%) als auch eigene Mitfahrtesuche (29,9 %) zu erstellen (53,8 % für eine Mischung aus beidem). Dabei wird die Möglichkeit der separaten Eingabe von Gesuchen als sinnvoll erachtet (96,5 %). Weiterhin wird die Implementierung eines Bewertungssystems mit überwiegender Mehrheit befürwortet (74,3 %).

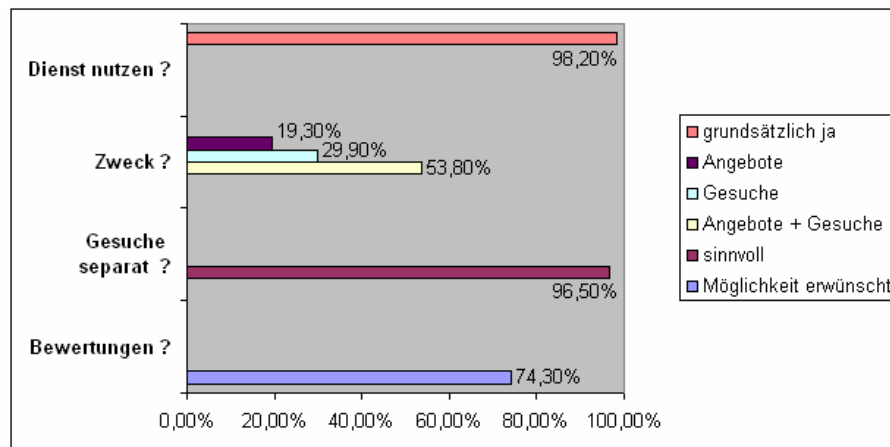


Abbildung 1-7 ausgewählte Umfrageergebnisse - allgemein (Microsoft Excel 2003)

² Die komplette Ausarbeitung zu der betreffenden Umfrage ist zu finden auf den Seiten 11-15 der Arbeit „Konzeption und Implementierung eines Mitfahrdienstes“ von Sandro Gerstenberger (2003)

Die folgenden zwei Abschnitte befassten sich jeweils mit detaillierten Charakteristiken der Funktionalität des zu erstellenden Dienstes.

Aus **Sicht des Fahrers** bewerteten 94,8 % der Befragten die Angabe von Zwischenhaltepunkten als nützlich. Bei der Frage zum Thema Fahrtkosten waren die gegebenen Antworten jedoch nicht so eindeutig. Etwa die Hälfte der Stimmen verteilte sich auf die beiden konkreten Optionen für einen Festbetrag pro Kilometer (25,1 %) oder pro Fahrt (26,3 %). Der mehrheitliche Rest (41,6 %) entschied sich für die individuelle Preisab-sprache als empfehlenswerteste Methode. Auch die Mitnahme von Gepäck sollte laut Abstimmung am besten individuell geklärt werden (81,9 %), wobei jedoch nur knapp fünf Prozent eine generelle Angabe zur Gepäckmitnahme allgemein für unnötig befanden. Fast die Hälfte der Befragten (48 %) sprach sich generell gegen das Rauchen im Auto aus, wobei 29,2 % diese Angelegenheit gerne über den Dienst regeln lassen würden. Nur 12,9 % finden diese Thematik unerheblich für einen Mitfahrdienst. Mit großer Mehrheit (88,9 %) befürworteten die Nutzer jedoch die Funktion der Mailbenachrichtigung (für Zusagen, Absagen, Änderungen etc.).

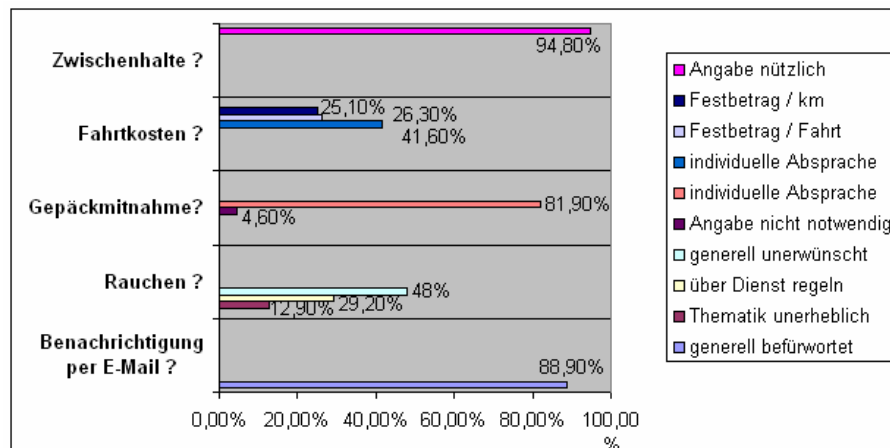


Abbildung 1-8 ausgewählte Umfrageergebnisse - Fahrersicht (Microsoft Excel 2003)

Für einen **Mitfahrer** wäre laut der Abstimmung eine Angabe zur Abfahrtszeit sinnvoll (nur 1,8 % waren dagegen), wobei die individuelle Absprache auch hier bevorzugt wird (72,5 %). Dies gilt auch für den Treffpunkt (71,7 %). Bei den Angaben zur Problematik Rauchen im Fahrzeug hat sich die Mehrheit für die Option „über Dienst regeln“ entschieden (59,0 %), wobei die restlichen Stimmen auf die Auswahlmöglichkeiten „ja“ (19,9 %) und „für Dienst unerheblich“ (21,1 %) verteilt waren. Die prozentuale Verteilung zum Punkt „Fahrtkostenbeteiligung“ stimmt tendenziell mit der aus Sicht des Fahrers überein. Geteilter Meinung waren die Beteiligten Studenten über die Integrierung eines Profils zur näheren Beschreibung des Fahrers über zusätzliche persönliche Angaben. Mit 51,5 % gab es jedoch einen leichten Vorsprung der Befürworter einer solchen Funktion.

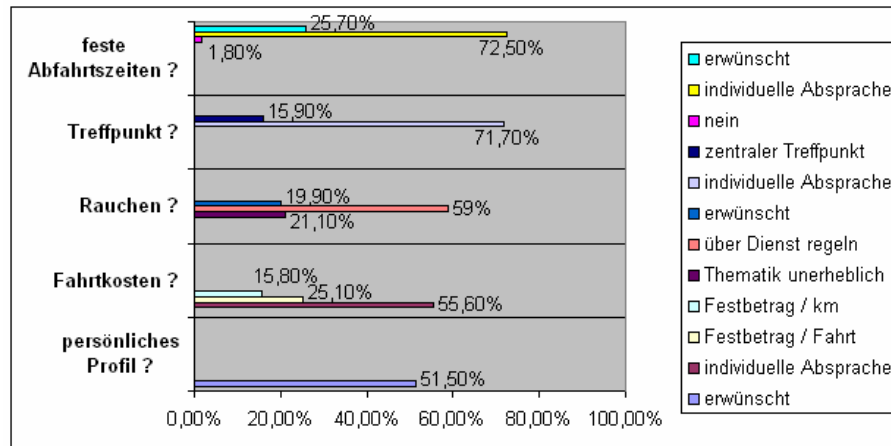


Abbildung 1-9 ausgewählte Umfrageergebnisse - Mitfahrersicht (Microsoft Excel 2003)

Insgesamt bestätigt diese Umfrage weitestgehend die Tendenzen, welche sich schon aus der Untersuchung einiger bereits existierender Dienste dieser Art ergeben haben. Erwartungsgemäß gab es leichte Abweichungen an einigen Stellen, ohne jedoch in grundlegend neuen Erkenntnissen zu resultieren. Deswegen hätte eine aktuelle Umfrage wahrscheinlich keine wesentlichen Änderungen ergeben. Ein paar Differenzen wären normal gewesen, aber diese hätten sich auch schon durch eine leichte Schwankung in der Teilnehmerzahl ergeben können.

1.3 Allgemeine Anforderungen an einen Mitfahrdienst

Nach der Analyse der Funktionalität bereits bestehender Mitfahrdienste und der persönlichen Ansichten der zukünftigen Nutzergruppe zum Thema, sollen nun zusammenfassend die grundlegendsten erforderlichen Merkmale eines zu erstellenden Dienstes dieser Art abgeleitet werden.

Als erstes wesentliches Kriterium steht bei allen webbasierten Mitfahrdiensten die Notwendigkeit der Registrierung, um die angebotenen Funktionsmerkmale voll nutzen zu können. Weil der hier zu entwickelnde Mitfahrservice in den Intranetbereich der Hochschulwebsite integriert werden soll, wird die grundlegende Autorisierung schon automatisch über den Intranet-Login geregelt (Zugangsberechtigungen, personalisierte Anzeigen etc.). Es hat sich gezeigt, dass die Angabe von ausgewählten persönlichen Details hilfreich ist, beispielsweise zur besseren Erkennung von unbekannten Nutzern oder um diese vertrauter wirken zu lassen. Jedoch sollte die Eingabe jeglicher im Dienst verwendeter Informationen logisch abgetrennt werden, um ein Formular im Sinne der Nutzerfreundlichkeit nicht zu überladen wirken zu lassen. Deshalb würde sich ein eigenständiges Formular für personenbezogene Angaben als nützlich erweisen. Dort könnte beispielsweise die zur Kontaktaufnahme benötigte E-Mail Adresse frei wählbar eingetragen werden (anstatt die eigene Standard Mailadresse der Hochschule verwenden zu müssen). Da diese Information als obligatorisch zu betrachten ist, macht es Sinn die Komplettierung aller Pflichteingaben ähnlich der Registrierung bei einem externen Dienstanbieter als Voraussetzung für die vollständige Dienstnutzung zu betrachten. So wäre die komplette Autorisierung eine Kombination aus der Intranetanmeldung und dem persönlichen Profileintrag.

Die für einen Mitfahrdienst notwendigen Funktionsgruppen können wie folgt aufgeteilt werden:

- *Eintragung von Inseraten*
 - Angebote
 - Gesuche
- *Verwaltung von Inseraten*
 - übersichtliche Auflistung der vorgenommenen Einträge mit optionaler Detailansicht zum jeweiligen Eintrag
 - Bearbeiten der Einträge
 - Löschen der Einträge
- *Suche in bestehenden Inseraten*
 - einfache Suche mit wenigen Parametern für schnellen Einstieg in die Nutzung des Dienstes
 - Detailsuche mit optionaler Parameterauswahl für eine gründlichere Selektion der Einträge
 - übersichtliche Auflistung der Suchergebnisse mit optionaler Detailansicht zum jeweiligen Eintrag
- *Buchen von Inseraten*
 - eindeutige Nutzer-Nutzer Zuordnung für die Bewertung
 - Verwaltung der verfügbaren Plätze
- *Verwaltung der Buchungen*
 - Übersichtliche Auflistung der eigenen und bei einem vorgenommenen Buchungen mit optionaler Detailansicht zum jeweiligen Eintrag
 - persönliche Details zum betreffenden Nutzer abrufen
 - Bearbeiten der eigenen Buchungen
 - eigene Buchungen löschen
- *Bewertungen vornehmen*
 - Nach der Fahrt den betreffenden Fahrer bzw. Mitfahrer bewerten
- *Bewertungen verwalten*
 - Übersichtliche Auflistung der erhaltenen und abgegebenen Bewertungen mit optionaler Detailansicht zum jeweiligen Eintrag

Die wichtigste Funktion des Dienstes ist selbstverständlich die Möglichkeit eigene **Inserate einzutragen**. Dabei erweist sich eine Trennung zwischen Angeboten und Gesuchen als sinnvoll. So können Personen, die kein passendes Angebot gefunden haben, ein eigenes Mitfahrge such erstellen, und Fahrer können so unmittelbar passende Mitfahrer finden. Oft haben vor allem regelmäßige Fahrer noch Plätze frei, ohne aber unbedingt gleich ein eigenes Angebot erstellen zu wollen. Eingetragene Gesuche könnten so die Vermittlungschancen erhöhen, weil dann ein zusätzlicher Anreiz zur Beteiligung geboten wird. Weiterhin würde sich eine Option für regelmäßige Inserate als nützlich darstellen, da viele potentielle Nutzer der Zielgruppe dieses Mitfahrdienstes in einem regelmäßigen Rhythmus zwischen Wohnort und Hochschule pendeln, und eine klare Kennzeichnung der entsprechenden Angebote bzw. Gesuche würde die Nutzerfreundlichkeit erhöhen.

Bei der Recherche hat sich außerdem gezeigt, dass die Angabe von Zwischenstationen bei Angeboten sehr hilfreich ist. Vor allem bei längeren Strecken gibt es viele potentielle Zwischenhaltepunkte, die relativ problemlos angefahren werden könnten. Durch die explizite Erwähnung derselbigen würden sich die Vermittlungschancen wesentlich steigern lassen, denn der Mitfahrdienst wäre so für eine größere Anzahl von Personen nutzbar. Dabei sollte die Suche diese Orte auch als mögliche Start- und Zielorte erkennen.

Weitere zusätzliche Details, um die Fahrt oder den entsprechenden Nutzer näher zu beschreiben, geben einen noch besseren Einblick bzw. Gesamteindruck und machen den Dienst noch attraktiver. Zu den sinnvollen Informationen dieser Art zählen beispielsweise die Angabe der Startzeit einer Fahrt, eine ungefähre preisliche Vorstellung oder ein konkreter Treffpunkt. Wie vorher schon erwähnt, wären auch Einzelheiten zur Person selbst, oder dem eigenen Fahrzeug denkbar. Jedoch hat sich eben auch gezeigt, dass eine übersichtliche Eingabemaske, für die jeweiligen Angebote bzw. Gesuche, definitiv einen positiven Einfluss auf die Handhabung eines Mitfahrdienstes hat. Deswegen sollten bestimmte logische Bereiche separat gehalten werden, um einen klaren Ablauf zu ermöglichen, der an keiner Stelle zu überladen wirkt.

Eine weitere elementare Funktion des Dienstes ist die **Suchmöglichkeit**. Die Inserate nur über so etwas wie eine Pinnwand zur Verfügung zu stellen wäre auf Dauer und mit wachsendem Nutzerumfang einfach zu unübersichtlich, und nicht wirklich ein Fortschritt gegenüber den bisherigen Möglichkeiten. Die grundlegendsten Parameter sind dabei „Ort“ und/oder „Datum“. Dadurch könnten die Inserate bereits ausreichend eingegrenzt werden. Für genauere Suchergebnisse sollte es eine Detailsuche mit zusätzlich möglichen Unterscheidungskriterien geben, was vor allem bei einer großen Anzahl von verfügbaren Inseraten an Wichtigkeit gewinnen würde um eine gründlichere Recherche zu ermöglichen.

Hat ein Nutzer eine passende Fahrt für sich gefunden, folgt im Allgemeinen als nächster Schritt die **Kontaktaufnahme** mit dem Inserenten. Dies wäre ein Grund für die Bedingung, mindestens die E-Mail Adresse hinterlegen zu müssen. Optional wäre auch die Angabe von Telefonnummern denkbar. Eine weitere Einsatzmöglichkeit der eingetragenen Mailadresse könnte ein automatisierter Versand von **Benachrichtigungsmittlungen** sein, etwa für den Fall einer vorgenommenen Buchung, der Absage oder Änderung einer solchen, oder der Änderung des Inserats durch den Urheber.

Jedoch ohne die Existenz eines **Buchungssystems** könnte sich ein potentieller Kunde bei der Recherche nie sicher sein, ob überhaupt noch Plätze für eine bestimmte Fahrt zur Verfügung stehen. Eine simple Platzverwaltung würde hierbei schon ausreichen,

damit die Notwendigkeit entfällt, unbedingt direkt Kontakt mit dem Inserenten aufnehmen zu müssen um dann erst darüber Auskunft zu erhalten. Die Anzahl der (noch) verfügbaren Plätze würde dann unmittelbar zur Verfügung stehen.

Ein weiterer Vorteil eines solchen Buchungssystems wäre eine schon eingebaute Sicherheitsoption, welche die grundlegendste Voraussetzung für die Implementierung eines Bewertungssystems darstellt. Denn diese Funktionalität macht nur Sinn, wenn ein Missbrauchsschutz existiert. Durch die personenbezogene Buchung würde hierbei schon eine feste Nutzer-Nutzer Zuordnung vorliegen, wobei immer nur die zwei jeweils beteiligten Personen eine gegenseitige Bewertung vornehmen dürften. Mit großer Mehrheit wurde während der hochschulinternen Umfrage der Einsatz einer **Bewertungsfunktion** befürwortet. Damit kann sich der interessierte Nutzer beispielsweise über die Zuverlässigkeit anderer Personen informieren, basierend auf Erfahrungswerten bisheriger Fahrer bzw. Mitfahrer.

Die Untersuchung der bereits etablierten webbasierten Mitfahrdienste hat gezeigt, dass sowohl Buchungs-, als auch Bewertungsmechanismen entweder gar nicht implementiert sind, oder Defizite in der Umsetzung aufweisen - die Handhabung sollte auf jeden Fall unkompliziert und eindeutig sein, um von den Nutzern überhaupt angenommen zu werden. Aber vor allem die Bewertung von Nutzern könnte sich auf Dauer als praktikabel erweisen, da es beispielsweise immer ein paar „schwarze Schafe“ zu geben scheint oder mit wachsender Anzahl an Nutzern natürlich immer mehr unbekannte Namen dabei sind. Und da ein Buchungssystem wie vorher schon erwähnt generell sinnvoll und in Verbindung mit der Bewertungsfunktion auch noch zweckmäßig ist, bietet es sich regelrecht an, dies bei einer Neuentwicklung eines Mitfahrdienstes auch mit zu realisieren.

Die aus den aktiven Hauptfunktionen des Dienstes resultierenden Einträge – Angebote/Gesuche, Buchungen, Bewertungen - sollten jeweils anhand einer übersichtlichen Auflistung verwaltet und eingesehen werden können, wobei eine logische Anordnung der Funktionskomponenten (Bearbeiten, Löschen etc.) die Nutzung wesentlich erleichtert.

Hier eine **Zusammenfassung wichtiger Kriterien** des Dienstes:

- Autorisierung des Nutzers
- Inserate als Angebote oder Gesuche definierbar
- einmalige und regelmäßige Inserate möglich
- Zwischenhaltepunkte bei Angeboten bestimmbar
- optionale Angabe zusätzlicher Details zur näheren Beschreibung
- Angaben zu Person oder Fahrzeug über separate Eingabemaske
- variable Suchmöglichkeit – Aufteilung in einfache Suche und Detailsuche
- Platzverwaltung durch Buchungsfunktion
- Bewertungssystem als Möglichkeit der Auskunft über andere Nutzer

2 Dienstentwurf

2.1 Umgebung des Dienstes – das Studentenportal

Das Studentenportal der Website der Hochschule Mittweida kann als eine Art Dienstportal betrachtet werden. Verschiedenste Serviceangebote rund um den Bedarf eines Studenten dieser Bildungseinrichtung sollen hier zentral versammelt zugänglich gemacht werden. Viele Leistungen dieser Art sind schon fertig implementiert und einige, wie auch der in dieser Bachelorarbeit behandelte Mitfahrdienst, befinden sich noch in der Entwicklung. Sowohl hochschulinterne Aspekte, als auch Dinge aus dem privaten Bereich werden hier abgedeckt.

Einige dieser Funktionsmerkmale (wie etwa Einschreibungen, Stundenplan, Notenanzeige etc.) existieren bereits seit geraumer Zeit und waren bisher nur verstreut entweder direkt im Intranetbereich der Website, oder umständlich über mehrere Seiten verteilt im allgemeinen Teil zu finden (teilweise sogar über verschiedene Wege zu erreichen). So konnte es durchaus vorkommen, dass ein Student sich dort auf der Suche nach einer bestimmten Auskunft regelrecht „verlaufen“ hat. Ein deutliches Indiz für die Notwendigkeit einer klaren und eindeutigen Struktur bzw. Navigation. Außerdem gab es bisher fast nur Dienstleistungen aus dem direkten Aufgaben- und Zuständigkeitsbereich der Hochschule. Somit ist es auch ein Ziel, mehr Dienste für den Studenten selbst zu realisieren, beispielsweise so etwas wie einen Wohnungsmarkt oder eben einen Mitfahrdienst. Anregungen für sinnvolle Ergänzungen könnten beispielsweise die Pinnwände in der Mensa oder vorgenommene Einträge im Forum geben.



Abbildung 2-1 Screenshot Startseite Studentenportal der Hochschulwebsite

Die **Struktur des Portals** ist derzeit thematisch in folgende Gruppen aufgeteilt:

- **Studienangebote**

Hier können sich angehende, sowie bereits eingeschriebene Studenten über die einzelnen an der Hochschule angebotenen Studiengänge informieren. Des Weiteren sind hier alle relevanten Informationen zur Bewerbung für diese Studienplätze zu finden (wie etwa Termine, Zulassungsvoraussetzungen, Immatrikulation etc.).

- **Studium und Alltag**

In diesem Bereich findet man eine große Fülle an Dingen aus dem studentischen Alltag. Als Kernstück kann dabei wohl der sogenannte „WegWeiser“ bezeichnet werden, ein „Portal im Portal“. Hier bietet sich dem Studenten eine Übersicht mit entsprechenden weiterführenden Informationen zu allen relevanten Bereichen des Studiums selbst (Immatrikulation, Prüfungen, Praktikum, Abschlussarbeit etc.), aber auch zu den einzelnen Hochschulinstitutionen (Hochschule, Bibliothek, Netzzentrum, Studentensekretariat etc.) und zu Themen des studentischen Lebens (Wohnung, Finanzierung, Freizeit und Sport etc.).

Die weiteren Bestandteile des Bereichs „Studium und Alltag“ beinhalten den direkten Zugriff auf den Stundenplan und die Mitgliederliste der eigenen Seminargruppe, personenbezogene Einschreibungsoptionen und die Möglichkeit der Noteneinsicht.

- **Groupware**

Hier findet man den persönlichen Terminkalender und die Beratungsplanung. Vor allem das Terminmanagement bietet eine gelungene Option alle festen Zeitpunkte und Verabredungen im Blick zu behalten. Lehrveranstaltungen, Prüfungen und separat getroffene Terminvorschläge können über diese Funktion verwaltet und beispielsweise auch mit Outlook synchronisiert werden, wobei auch eigene Eintragungen möglich sind.

- **Dienstleistungen**

Der Mitfahrservice ist in diesem Abschnitt zu finden. Aber auch ein Wohnungsmarkt wurde integriert. Da es ständig Bedarf an kostengünstigem Wohnraum gibt, aber bisher keine zentrale Anlaufstelle in diesem Bereich speziell für die Studenten der Hochschule Mittweida existierte, ist dieser Service ein weiterer Beleg für den Nutzen des Studentenportals. Sowohl Studenten selbst, als auch externe Anbieter haben die Möglichkeit, hier ihre Angebote in eine Datenbank einzutragen. Über eine Suche erfolgt dann bei Bedarf der Zugriff auf diese Einträge. Ein konkreter Wohnraum kann dabei mit zahlreichen Details beschrieben werden, von Standard Informationen wie Anschrift, Größe und Mietpreis bis hin zu Ausstattungs- und Einrichtungsmerkmalen wie Küchen- und Badeinrichtung, Möblierung oder dem Internetanschluss. Die eigenen Angebote können übersichtlich verwaltet, und fremde Angebote vorgemerkt werden.

Weiterhin in diesem Abschnitt aufgeführt sind wertvolle Hinweise, um an kostenlose oder kostengünstige Studentenlizenzen für wichtige Softwareprodukte großer Anbieter heranzukommen. Oft ist es nicht möglich oder praktikabel, alle zu erledigenden praktischen Aufgaben vor Ort in der Hochschule zu erledigen. Des-

wegen sind solche speziellen Tarife für den durchschnittlichen Studenten die einzige legale Möglichkeit diese Produkte trotzdem nutzen zu können.

- ***Wichtige Links / Organisationen, Gremien, Vereine für Studenten***

Alle über diesen Abschnitt zugänglichen Informationen sind kein direkter Bestandteil des Studentenportals oder der allgemeinen Hochschulwebsite selbst - ausschließlich externe Quellen wurden hier verlinkt. Jedoch handelt es sich jeweils um hochschulzugehörige oder -verbundene Institutionen, wie beispielsweise den Studentenrat, das Studentenwerk, den CosmopolitanClub etc. Diese Einrichtungen mit separaten Webauftritten bieten wichtige Informationen bzw. Anlaufstellen für die Studenten und gehören deshalb genauso zu den Auflistungen im Studentenportal wie alle anderen, internen Informationsquellen.

Abgerundet wird das Portal durch einen Campusplan und den Zugang zum Intranetbereich der Hochschulwebsite.

2.2 Projektstand bei Übernahme

2.2.1 Kritik

Beim ersten Betrachten des Mitfahrdienstes im ursprünglichen Zustand fiel direkt auf, dass ein Großteil der einzelnen Funktionalitäten nur eingeschränkt, oder gar nicht nutzungsfähig war. Daraus ergab sich der große Nachteil, den Dienst selbst nicht richtig ausprobieren zu können. Da jedoch letztendlich sowieso eine Neuentwicklung angestrebt wurde, war es somit verstärkt erforderlich, eigene Gedanken und Ideen zum konkreten Ablauf und den internen Zusammenhängen zu entwickeln. Als Abhilfe konnten allerdings die Dokumentation zum Dienst³ oder auch die Quelltexte selbst als Studienobjekte zu Rate gezogen werden. Die Ursachen des fehlerhaften Verhaltens lagen wohl hauptsächlich bei den inzwischen überholten Einbindungen der jeweiligen Funktionsbibliotheken oder den veralteten Versionen der Bibliotheken selbst.

Abgesehen von den praktischen Gründen einer Neukonzipierung des Mitfahrdienstes gibt es auch inhaltliche Aspekte der vorherigen Umsetzung, die einerseits objektiv, andererseits aufgrund individueller Ansichten nicht optimal erscheinen. Ein paar der zentralen Auffälligkeiten sollen hierbei kurz angesprochen werden.

- **Quelltexte**

Schon der Quelltext selbst ist zwar oberflächlich betrachtet ordnungsgemäß gegliedert und formatiert, aber die tiefer geschachtelten Ebenen sind meist schwierig durchzusehen bzw. zu trennen. Eine einfache Änderung von wenigen Variablen erweist sich teilweise schon als äußerst schwierig, vor allem für jemanden der die Programmierung nicht selbst vorgenommen hat. Auf dieser Basis das gesamte Projekt bearbeiten zu müssen wäre auf Dauer viel zu umständlich. Eine übersichtlichere Trennung bzw. Formatierung der einzelnen Ebenen würde die Programmierung nicht nur für den Urheber selbst, sondern auch für fremde Personen erleichtern, was zwingend erforderlich ist wenn beispielsweise das Projekt später unabhängig vom ursprünglichen Verfasser weitergeführt werden soll.

- **Grafisches Interface**

- *Formulardesign*

Aber auch beim äußeren Design, dem User Interface, gibt es ein paar Schwächen. Auffällig ist zum Beispiel die Struktur der Formulare, bzw. deren Formatierung. Es fehlt an manchen Stellen die klare, lineare Anordnung. Verschiedene Elemente wie Eingabefelder und Buttons sind optisch verschoben, ohne absolute Platzierung. Schon der Einsatz von, möglicherweise zusätzlichen, Tabellen in diversen Abschnitten hätte für ein höheres Maß an Strukturierung gesorgt. Dafür gibt es aber zum Zeitpunkt des in dieser Arbeit entwickelten Dienstes genauere Vorgaben zur Anfertigung von Formularseiten, die als Ausgangspunkt genutzt werden können. Vorher war eine freiere Gestaltung möglich, was im Allgemeinen sowohl Vor-, als auch Nachteile mit sich bringt. So ergab sich teilweise das Problem (oder die Gefahr) einer uneinheitlichen Gesamtgestaltung, gerade auch im Bezug auf das allgemeine Design der

³ Sandro Gerstenberger: Konzeption und Implementierung eines Mitfahrdienstes, 2003

Hochschulwebsite. Korrespondierende Gestaltungsmerkmale und -elemente einzuarbeiten stellt demnach ein Hauptanliegen bei der Neugestaltung des Mitfahrdienstes dar.

- *Formularinhalt*

Jedoch auch der inhaltliche Aufbau des grafischen Interface der ursprünglichen Version bedarf einer leichten Überarbeitung, um eine optimierte Nutzung zu ermöglichen. Das heißt, ein Formular sollte in logische Segmente aufgeteilt sein und so weit wie möglich keine unnötigen Bedienelemente und Eingaben enthalten, um den Nutzer nicht abzuschrecken oder zu überfordern.

- **Übersicht/Verwaltung**

Im selben Problemkreis liegen auch die jeweiligen Übersichtsseiten. Im Grunde ist dort inhaltlich auf die vorherige Realisierung bezogen nicht wirklich etwas falsch gemacht worden. Jedoch wäre eine einheitliche äußerliche Anpassung an aktuelle Gegebenheiten in jedem Fall nötig, und in diesem Zug ergibt sich dann auch die Chance, den konkreten Aufbau etwas zu modifizieren.

Was die eigentlichen Funktionsgruppen im Speziellen angeht, gibt es an ein paar Einzelheiten mit Abänderungsbedarf. Manche Features erscheinen dabei innerhalb der Neuentwicklung in fast komplett anderer Form, oder sie wurden aus einer anderen Sichtweise bzw. Motivation heraus entworfen. Dabei sollte festgestellt werden, dass die grundlegende Funktionalität in der ursprünglichen Version des Mitfahrdienstes gewährleistet war und einige der späteren Änderungen mehr aufgrund individueller Ansichten entstanden sind, teilweise ohne der zwingenden Notwendigkeit etwa hinsichtlich fehlerhafter Umsetzungen vorgehen zu müssen. Deswegen sollen an dieser Stelle nur ein paar Beispiele aus diesem Bereich zur besseren Verdeutlichung erwähnt werden.

- **Persönliche Details**

Persönliche Informationen waren vorher nur als optionale Angaben definiert, und es bestand keine klare Trennung zwischen den Daten des Inserats und der Person selbst. Als Weiterentwicklung ist deshalb in der neuen Fassung ein eigenständiges persönliches Profil, teils mit Pflichteingaben vorgesehen. Dabei sind alle Details zu einem Nutzer immer separat von einem konkreten Angebot oder Gesuch gehalten.

- **Buchungsfunktion**

- *Buchung und Gesuche*

Die Buchungsfunktion wurde in der ursprünglichen Version eher simpel gehalten, und war beispielsweise nur für Angebote ausgelegt. An sich ist das völlig ausreichend, um diese Funktionalität grundlegend logisch zu realisieren (ein Gesuch betrifft ja nur einen Platz, weshalb eine reine Platzverwaltung hierbei an sich nicht nötig wäre). Bezieht man jedoch auch die Gesuche mit ein, ergeben sich gewisse Vorteile. Insgesamt wirkt die Buchungsfunktion damit durchgängig einheitlicher für alle Inserate. Außerdem erhält dann auch der Urheber des Gesuchs eine positive Bestätigung bei konkretem Interesse eines anderen Nutzers und sein Inserat wird automatisch passiv geschaltet (was vorher selbstständig erledigt werden musste). Darüber hinaus können in

der entsprechenden Übersicht dann alle Aspekte der Nutzung verwaltet werden (und nicht nur ein Teil dessen). Denn eine Fahrtenverwaltung die beispielsweise die Hälfte aller Fahrten mit eigener Beteiligung gar nicht berücksichtigt - weil über Gesuche zustande gekommen - erscheint doch etwas halbherzig. Ein weiterer Kritikpunkt besteht in der Tatsache, dass wenn ein Gesuch nicht explizit gebucht werden kann, auch keine Bewertung dieser Fahrt möglich wäre. Obwohl die Buchung bei Gesuchen nicht zur Platzverwaltung benötigt wird, so ist diese doch zumindest für die Möglichkeit der späteren Bewertung essentiell erforderlich.

- *Buchung und Zeitparameter*

Weiterhin stellt sich eine Buchungsfunktion mit Platzverwaltung, aber ohne Differenzierung des Zeitpunkts (wie in der ursprünglichen Version vorhanden) unter Umständen als ineffizient dar. Bei einmaligen Fahrten ist diese Unterscheidung unerheblich. Da aber auch regelmäßige Angebote zugelassen sind, wären die gebuchten Plätze in dem Fall für den gesamten Gültigkeitszeitraum belegt. Und nur weil eine angebotene Fahrt beispielsweise als regelmäßig werktags definiert ist, bedeutet das nicht, dass beim Inserenten des Angebots auch nur automatisch Buchungen mit exakt dieser Frequenz akzeptiert werden oder zustande kommen können. Einzelfahrt- oder Einzeltagsbuchungen sollten hierbei normalerweise immer möglich sein. Im restlichen Gültigkeitszeitraum könnten dann noch Fahrten vergeben bzw. wahrgenommen werden, und durch die Verwendung des entsprechenden Zeitparameters in der Platzverwaltung ist dies für den interessierten Nutzer auch ersichtlich.

Die meisten dieser inhaltlichen Kritikpunkte bieten sinnvolle Ansätze für Änderungen die mehr im Sinne einer Verbesserung bzw. Weiterentwicklung zu verstehen sind. Andere Details, wie etwa das grafische Interface oder auch die im Dienst verwendeten Quelltexte, sind dagegen Elemente bei denen schon aufgrund des langen zeitlichen Abstands beider Arbeiten oder des Bearbeiterwechsels selbst Handlungsbedarf besteht.

2.2.2 Fazit

Von Juni bis November 2003 wurde im Rahmen der Bachelorarbeit von Herrn Sandro Gerstenberger bereits schon einmal ein Mitfahrservice für die Hochschule Mittweida konzipiert und auf Basis von PHP und HTML programmiert. Dieser Dienst ist strukturell betrachtet dem hier vorgestellten sehr ähnlich, was aufgrund der Mindestanforderungen betreffend grundlegende Funktionalität eines Mitfahrdienstes auch nicht anders möglich ist. Die wesentlichen Funktionsgruppen - wie etwa Inserate eintragen, Buchungen vornehmen, Fahrer bewerten – sind genauso wie in dieser neu entworfenen auch in der ursprünglichen Version vorhanden. Jedoch unterscheiden sich letztendlich beide Realisierungen sowohl in den jeweiligen Details ihrer Umsetzung als auch im optischen Erscheinungsbild.

Zum damaligen Zeitpunkt kam der von Herrn Gerstenberger entwickelte Dienst noch nicht zum Einsatz, und das Studentenportal existierte da nur als Entwurf ohne konkrete praktische Verwirklichung. Die Fertigstellung dieser ursprünglichen Fassung war zum Zeitpunkt des Bearbeitungsbeginns der hier behandelten Variante fast vier Jahre her, und das ist ein Grund warum die Programmierung im vorliegenden Zustand nicht genutzt

werden kann. Allein dieser lange Zeitabstand würde eine Überarbeitung rechtfertigen, aber entscheidend ist auch die Tatsache dass aufgrund neuer Umgebungsparameter beispielsweise das Design angepasst werden muss, oder auch dass neue bzw. verbesserte globale Funktionen/Features vorliegen und deren Einarbeitung im Bezug auf eine einheitliche Verwendung nötig ist. Auch die persönlichen Auffassungen und Änderungswünsche zum Thema müssten in die ja eigentlich schon fertige Umsetzung integriert werden. Deshalb stellt sich letztendlich eine Neuentwicklung von Grund auf als praktisch sinnvoll heraus. Die Einarbeitung in so ein umfangreiches Programmierprojekt und das spezifische Einpassen der verschiedenen Änderungen ist an sich schon mit einem gewissen Zeitaufwand verbunden. Deshalb kann diese Zeit auch genutzt werden, um ein eigenes Projekt anzugehen (bei dem hier zu erwartenden Projektvolumen ist dieser Schritt praktisch noch als sinnvoll zu erachten). Dadurch verstärkt sich außerdem der Lerneffekt und man erlangt ein besseres Verständnis für das ganze Projekt an sich. Im Endeffekt werden sich einige Aspekte der aktuellen Programmierung sowohl an der ursprünglichen Fassung, als auch an weiteren Komponenten des Studentenportals, beispielsweise dem Wohnungsmarkt, positiv orientieren.

2.3 Spezifikation des Dienstes

Im folgenden Abschnitt sollen nun die einzelnen Funktionsgruppen der Neuentwicklung des Mitfahrdienstes im Detail besprochen werden. Die gewonnenen Erkenntnisse aus den allgemeinen Anforderungen an einen solchen Service werden hierbei in die Überlegungen einfließen und als Basis dienen.

2.3.1 Grundkonzept und Struktur

Das grundlegende Nutzungsmodell ist recht simpel – Inserate können in eine Datenbank eingetragen und damit anderen Nutzern orts- und zeitunabhängig über die Hochschulwebsite zugänglich gemacht werden. Die Eingabe erfolgt mittels eines übersichtlichen Formulars, wobei prinzipiell zwischen den beiden Inseratstypen „Angebot“ und „Gesuch“ gewählt werden muss. Verwaltet, d.h. geändert bzw. gelöscht, werden die Einträge anhand einer anwenderfreundlichen Übersichtsseite. Eine Suchfunktion ermöglicht unterdessen das zielgenaue Auffinden passender Inserate, welche bei Interesse gebucht werden können - um einen oder mehrere Plätze fest zu reservieren und nach der Fahrt die gegenseitige Bewertung zu ermöglichen. Sowohl die Buchungen (Änderungen, Absagen und Fremdbuchungen), als auch die Bewertungen (abgegebene, erhaltene und offene) sind unter Einsatz personalisierter Übersichten komfortabel administrierbar.

Dieser Hauptumfang wird noch durch zusätzliche Features ergänzt, um die Funktionalität zu erhöhen und den Mitfahrdienst Nutzerfreundlicher zu gestalten. So gibt es beispielsweise separate Profile für Informationen zur Person und zum Fahrzeug. Wobei grundlegende Angaben wie Name und E-Mail Adresse als Pflicht zu betrachten, und als Mindestvoraussetzung zur vollständigen Nutzung des Dienstes erforderlich sind. Die weiterführenden Angaben können jederzeit abgerufen werden, und beinhalten unter anderem auch die Bewertungsdetails eines Nutzers und relevante Kontaktinformationen.

Die Startseite des Dienstes fungiert insgesamt als Ausgangspunkt für dessen Verwendung. Alle Funktionen sind von hier aus zugänglich, wobei eine logische Anordnung mit Hilfe der Unterteilung in einzelne Funktionsgruppen vorliegt.

Die Aufteilung ergibt sich dabei wie folgt:

- *Angebote/Gesuche finden & buchen*
 - Einfache Suche (Ort/Datum)
 - Detailsuche mit zusätzlichen, meist optionalen Suchkriterien
 - Unmittelbare Anzeige aller eingetragenen und gültigen Angebote bzw. Gesuche
 - über Detailansicht konkrete Fahrt buchen
- *Angebote/Gesuche erstellen & verwalten*
 - Inserate eintragen
 - Inserate verwalten
- *Buchungen verwalten*
 - Buchungsübersicht
- *Bewertungen abgeben & verwalten*
 - Bewertungsübersicht
 - über entsprechende Verwaltungsoption konkrete Fahrt bewerten
- *Profile*
 - persönliches Profil eintragen (enthält relevante Pflichtangaben)
 - Fahrzeugprofil eintragen (optional)
 - eigene Profilansicht betrachten
 - eigenständige Suche zum Auffinden der separaten Profile eingetragener Nutzer

2.3.2 Eintragung/Verwaltung von Inseraten

Am Anfang der Dienstnutzung steht die Eintragung von Inseraten. Entweder weil man freie Plätze im eigenen Fahrzeug zur Verfügung stellen kann, oder da (noch) kein passendes Angebot vorhanden ist, und man seinen Fahrtenbedarf anzeigen möchte.

Dafür steht dem Nutzer ein Formular zur Verfügung, welches sowohl für Angebote, als auch für Gesuche genutzt werden kann. Eine Auswahl per Radio-Button und die Spalte „Art“ in der entsprechenden Tabelle der Datenbank genügen, um die vormals zwei Eingabeformulare zusammenzufassen. Da sich Angebote und Gesuche inhaltlich kaum unterscheiden, reicht eine vereinheitlichte Eingabe vollkommen aus.

▪ Formular

Das Formular selbst ist in folgende inhaltliche Segmente unterteilt:

- Art
- Zeit
- Strecke
- Details
- Zusatzinfos

Zu allen Elementen ist ein kurzer Erklärungstext vorhanden (erscheint als *mouse-over* -Effekt des entsprechenden Begriffs). Die Pflichteingaben sind in Rot gekennzeichnet, und bei fehlenden Angaben wird der Nutzer durch die integrierte Eingabeprüfung darauf aufmerksam gemacht.

Im Bereich **Zeit** wird die Angabe von Frequenz, Datum und Uhrzeit verlangt. Die Auswahlkomponente der Frequenz ist ein Element, welches als standardisiertes Objekt zur Verfügung steht und deshalb unverändert übernommen wurde. Ob eine Fahrt einmalig oder regelmäßig (täglich, werktags, montags etc.) stattfinden soll, wird hier entschieden. Die Datumseingabe erfolgt per Hand oder bequem als Auswahl über die Kalenderoption. Soll ein einzelner regelmäßiger Wochentag gebucht werden, wird das Datum entsprechend auf den passenden Tag geprüft. Oft kann man sich weit im Voraus jedoch nicht auf eine konkrete Abfahrtszeit festlegen, oder man ist variabel was diesen Punkt angeht und die Absprache unter den Mitreisenden währe letztendlich entscheidend. Deswegen gibt es bei der Uhrzeit auch die Möglichkeit „VHS“ für Verhandlungssache zu wählen.

Der Abschnitt **Strecke** beinhaltet Start- und Zielort, sowie bei Angeboten die optionalen Parameter der Zwischenstationen. Die Zielgruppe dieses Dienstes sind die Studenten der Hochschule Mittweida, und diese bewegen sich überwiegend im näheren Umkreis, bzw. bewältigen eher kürzere Distanzen im Gegensatz zu beispielsweise deutschlandweiten Fahrten. Deshalb sollten maximal drei Zwischenhaltepunkte ausreichen. Außerdem wird die Verarbeitung der Daten so nicht unnötig verkompliziert.

Nach den grundlegenden Informationen folgen die **Details**, welche die Fahrt näher beschreiben. Als Pflichtangabe ist dabei die konkrete Anzahl der verfügbaren bzw. erforderlichen Plätze zu betrachten (ist diese Anzahl als Standard im Fahrzeugprofil festgelegt, wird sie hier automatisch übernommen). Ein fester Treffpunkt oder die Angabe einer bestimmten preislichen Vorstellung sind dagegen nicht zwingend notwendig, aber informativ.

Falls ein Nutzer dieses Dienstes nun beispielsweise täglich zwischen Wohnort und Hochschule pendelt, würde es sich ja anbieten, sowohl Hin-, als auch Rückfahrt einzutragen. Um nicht das gleiche Inserat praktisch nochmals eintippen zu müssen, kann bei jedem eigenständigen Eintrag die Rückfahrtsoption gewählt werden. In dem Fall wird man nach der Komplettierung des Formulars direkt auf die Eingabe der Rückfahrt weitergeleitet, wobei die relevanten Parameter bereits

in angepasster Form (Startort = Zielort etc.) ausgefüllt sind. Einzelne Angaben dürfen dabei nicht, oder nur eingeschränkt abgeändert werden (wie etwa die Art und die Frequenz). Ändert der Nutzer hierbei bestimmte kritische Parameter, wird er darauf aufmerksam gemacht und vor die Wahl gestellt, diese endgültig abzuspeichern (und damit den Status als Rückfahrt aufzugeben) oder die Änderungen noch rückgängig zu machen. Nur bei gültiger bzw. kompatibler Eingabe der Rückfahrt werden der Auswahlparameter und die zugehörigen Inseratskennungen letztendlich auch im Haupteintrag zugewiesen.

Abschließend können noch **zusätzliche Bemerkungen** abgegeben werden, beispielsweise um Details zu erwähnen, die nicht von den Formularparametern abgedeckt wurden.

- **Verwaltung**

Die Verwaltung der Inserate erfolgt mittels der Übersicht für Angebote und Gesuche. Alle Einträge sind entsprechend ihrer Art eingeordnet, wobei auch ein Abschnitt für vorgemerkte Fahrten, eine Art „Favoritenliste“, existiert. Die eigenen Inserate können geändert, gelöscht oder im Detail betrachtet werden. Weiterhin gibt es die Möglichkeit, Angebote oder Gesuche zu kopieren. Das heißt, ein neuer Eintrag kann direkt aus einem bestehenden erstellt werden – wenn etwa eine bestimmte Fahrt oder Strecke häufiger stattfindet, muss nicht jedes Mal der komplette Eintrag neu erstellt werden. Die Änderungen und Kopien erfolgen über dasselbe Formular wie die Eintragung der Inserate selbst; es werden nur die entsprechenden Daten in das Formular geladen.

- **Detailbetrachtung**

Um die eingetragenen Informationen auch für andere Nutzer zugänglich zu machen, werden die Daten aufgearbeitet und mittels einer einzelnen Detailseite zusammengefasst dargestellt. Die Optik erinnert an das vertraute Design des vorangegangenen Formulars, und auch die inhaltliche Struktur ist daran angelehnt. Lediglich der unterste Abschnitt wurde neu hinzugefügt. Dort findet sich der Link zum angepassten Buchungsformular und die Möglichkeit, das Inserat vorzumerken. Weiterhin kann der Nutzer über diese Seite die persönliche Profilkarte des Inserenten abrufen.

Betrachtet der Inserent die Detailseite seiner eigenen Eintragung, hat er außerdem die Möglichkeit, unter dem Punkt „bisherige Buchungen“ alle Buchungen speziell für dieses Inserat aufgelistet zu bekommen.

Die erfassten Daten zum Inserat werden im Sinne der Anschaulichkeit teilweise bearbeitet und entsprechend dargestellt. Verhandlungssache statt „VHS“, „keine Angabe“ wenn bestimmte Felder ausgelassen wurden, die Anzeige ob eine Rückfahrt eingetragen ist (inklusive Direktlink zum entsprechenden Inserat), Datumsanzeige inklusive Wochentag oder Anzeige des letzten Änderungszeitpunkts.

▪ Inserat Bearbeiten

Die Bearbeitung erfolgt wie schon erwähnt über dasselbe Formular wie die Eintragung des Inserats selbst, unter aufrufen der entsprechenden Informationen aus der Datenbank. Ohne bestehende Buchung kann das Inserat zu jedem Zeitpunkt beliebig modifiziert werden. Liegt jedoch für dieses Inserat bereits mindestens eine Buchung vor, sind uneingeschränkte Änderungen nur bei nicht kritischen Parametern möglich. Art, Startort und Zielort dürfen nicht, sowie Frequenz, Datum, Zwischenhaltepunkte und die Platzanzahl nur bedingt geändert werden. Dabei ergeben sich leichte Unterschiede zwischen Angeboten und Gesuchen:

- Angebote

- *Datum*

Bei einmaligen Fahrten muss das bisherige Datum bestehen bleiben. Das Startdatum jeglicher regelmäßiger Inserate kann vorverlegt werden, darf aber nicht nach dem ursprünglichen Beginn liegen.

- *Zwischenhaltepunkte*

Im Gegensatz zu Start- und Zielort können die Zwischenhaltepunkte ohne weiteres geändert werden. Hat ein Nutzer jedoch einen der betreffenden Orte explizit gebucht, ist dieser für Änderungen gesperrt.

- *Platzanzahl*

Modifiziert der Nutzer den Parameter der verfügbaren freien Sitzplätze, darf diese Anzahl nicht unterhalb der bereits gebuchten Menge liegen.

- *Frequenz*

In diesem Bereich können bei vorliegenden Buchungen ausschließlich Optionen gewählt werden, welche die ursprüngliche Frequenz einschließen. Das heißt: einmalige Fahrten müssen auch so belassen werden; alle regelmäßigen, einen Wochentag betreffenden Fahrten können auf täglich (Mo-So) oder werktags (Mo-Fr) erweitert werden, was auch auf den Wechsel von werktags auf täglich zutrifft.

- Gesuche

- *Datum*

Einzelfahrten dürfen nicht verlegt werden. Alle regelmäßig stattfindenden Inserate können bei Gesuchen nur nach hinten verlegt werden, anstatt wie bei Angeboten nach vorne.

- *Zwischenhaltepunkte*

Dieser Parameter steht bei Gesuchen nicht zur Verfügung.

- *Platzanzahl*

Hierbei darf die Zahl der benötigten Plätze nur reduziert, aber nicht erweitert werden.

- *Frequenz*

Die neu gewählte Option muss Teil der vorherig gültigen Frequenz sein. Das heißt, im Gegensatz zu den Angeboten ist hier nur eine Herabstufung möglich: täglich auf werktags; täglich/werktags auf entsprechende Einzeltage oder Einzelfahrten und einzelne Wochentage auf an diesem Tag stattfindende einmalige Fahrten.

Liegen die Änderungswünsche eines Nutzers außerhalb der definierten Grenzen, besteht die Möglichkeit, allen Buchenden automatisch eine Mitteilung zukommen zu lassen, in welcher die Änderungsabsicht angezeigt wird. So wissen die betreffenden Personen bescheid, und können sich auch selbst mit dem Inserenten in Verbindung zu setzen, um die Angelegenheit zu klären. Außerdem lässt sich zwar die Rückfahrtsoption in jedem Fall ändern, jedoch sollte beachtet werden, dass beispielsweise die Rückfahrt bei der Änderung von ja auf nein gelöscht wird, auch wenn diese schon gebucht wurde (analog zum normalen Löschen eines Inserats). Entsprechende Mitteilungsmails werden verschickt, und die betreffenden Buchungs- und temporären Bewertungseinträge gelöscht.

Die Rückfahrt selbst kann auch ohne vorliegende Buchung nur bedingt geändert werden (um noch als Rückfahrt gewertet werden zu können), da sie ja mit dem Haupteintrag korrespondieren muss. Folgende Einschränkungen gelten:

- *Datum*

Der Rückfahrtstermin darf nicht vor der Hinfahrt liegen.

- *Start-/Zielort, Zwischenhaltepunkte*

Diese Parameter dürfen bei der Rückfahrt nicht verändert werden, da es sich ja sonst nicht mehr um dieselbe Fahrt handeln würde.

- *Art*

Auch diese Option muss zur Hinfahrt übereinstimmend belassen werden.

- *Frequenz*

Auch in diesem Fall können Einzelfahrten nicht verändert werden. Für Hin- und Rückfahrt gibt es aber folgende kompatible Auswahlmöglichkeiten: täglich und werktags, jeweils aber nicht in Kombination mit einem einzelnen Wochentag; Einzeltage können jedoch beliebig untereinander kombiniert werden.

Im Gegensatz zu Inseraten mit bestehenden Buchungen können kritische Änderung in diesem Fall bei Bedarf trotzdem vorgenommen werden, jedoch geht unter den oben genannten Umständen der Status einer Rückfahrt verloren. Alle nicht kompatiblen Modifikationen werden dem Nutzer vor dem endgültigen Abspeichern mit einer Warnung angezeigt, und können an dem Punkt noch rückgängig gemacht werden. Sind Änderungen geplant, sollten diese immer zuerst im Haupteintrag (= Hinfahrt) vorgenommen werden.

Diese allgemeine Änderungsblockade ist in der Zuverlässigkeit des Dienstes begründet, in dem Sinne dass jede Nutzergruppe nur im Bezug auf ihre Einträge administrative Rechte besitzt. Das heißt in diesem Fall, eine bestehende Buchung währe letztendlich hinfällig, wenn das Inserat in gewissen Schlüsselpunkten geändert werden würde (der Eintrag an sich jedoch bestehen bleibt). Beim Löschen des Angebotes/Gesuches muss jedoch der Inserent das letzte Wort haben um seinen Eintrag im Notfall immer zurückziehen zu können.

- **Inserat löschen**

Beim Löschen müssen zwei Situationen unterschieden werden: Das Inserat besitzt keine Buchung, oder das Inserat wurde bereits gebucht. Ohne Buchung wird der Nutzer im Sinne einer Sicherheitsmaßnahme vor dem Löschen erst auf eine Art Vorschauseite weitergeleitet, wo er noch die Chance hat, sich anders zu entscheiden. Danach wird der bestehende Eintrag direkt aus der Datenbank entfernt. Liegt mindestens eine Buchung vor, wird der Nutzer auf der folgenden Vorschauseite mit einer Warnung darauf hingewiesen. Inserate können jedoch aus administrativen Gründen immer gelöscht werden, und in diesem Fall bekommt der buchende Nutzer zumindest eine entsprechende Mitteilung per Mail zugesandt. Entsprechende Buchungs- und temporäre Bewertungseinträge in der Datenbank werden dann zusätzlich zum Inserat selbst mit entfernt.

- **Persönliche Details**

Alle weiterführenden Details zur Person oder zum Fahrzeug werden über separate Formulare geregelt und müssen demnach nicht bei jedem einzelnen Inserat neu eingegeben werden. Die Angaben zu einem Nutzer werden auf einer separaten Profilseite aufgeführt. Diese Seite ist beispielsweise über die Inseratsdetails zugänglich („Inserat von“). Aber auch in den Buchungsdetails, Bewertungsdetails oder direkt über die Personensuche kann darauf zugegriffen werden.

2.3.3 Suchfunktion

Je größer die Anzahl der aktiv am Dienst beteiligten Personen, umso mehr Inserate befinden sich in der Datenbank. Ohne Suchmöglichkeit wäre dieser Service damit auch nur eine bessere Pinnwand, und würde keinen wirklichen Fortschritt zu den bisherigen Methoden darstellen. Deshalb stellt die Suche neben der Möglichkeit eigene Inserate einzutragen die wichtigste Komponente des Mitfahrdienstes dar.

Dabei sind in diesem Fall zwei Varianten implementiert worden: die einfache Suche und die Detailsuche. Oft wollen sich manche Nutzer nur auf die Schnelle über eventuelle Fahrten informieren, oder haben nur wenige konkrete Einzelheiten im Sinn, weswegen eine detaillierte Suche so nicht zwingend nötig wäre. Außerdem könnten die zahlreichen Auswahloptionen möglicherweise etwas abschreckend wirken, wenn man beispielsweise nur schnell in Erfahrung bringen will, welche Fahrten für den Ort Dresden eingetragen sind.

Zusätzlich zur Suche für Inserate gibt es auch die Möglichkeit, direkt nach den Profilen der eingetragenen Nutzer suchen.

- **einfache Suche**

Diese Methode ist direkt über die Startseite des Dienstes nutzbar. Hierbei stehen nur maximal zwei Parameter zur Filterung der Einträge zur Verfügung. Es kann nach dem Ort, dem Datum oder der Kombination aus Beidem gesucht werden (bleiben beide Textfelder leer, werden alle Inserate aufgelistet). Diese Variante ist hinreichend, um die Resultate in einem ausreichenden Maß einzugrenzen.

- **Detailsuche**

Hat ein Nutzer konkretere Vorstellungen was die Details der Fahrt angeht, oder sollen die Ergebnisse noch genauer voneinander differenziert werden, dann erweitert sich diese ausführlichere Suchmöglichkeit als praktisch sinnvolle Methode der Recherche. Von der Startseite aus gelangt man über den entsprechenden Link in das Suchformular. Dieses ist, in Anlehnung an das Formular zur Eingabe der Inserate, thematisch wie folgt gruppiert:

- Art
- Zeit
- Strecke
- Details

Demnach gibt es auch bei den implementierten Parametern Übereinstimmungen. Art, Frequenz, Datum, Uhrzeit, Startort und Zielort sind auch hier zu finden, teilweise jedoch mit für die Suche spezifischen Besonderheiten. So ist es möglich bei Datum und Uhrzeit einen Toleranzbereich zu definieren, mit Abweichungen der angegebenen Werte von bis zu fünf Tagen bzw. Stunden (plus/minus). Außerdem kann für den schnellen Zugriff bei den Orten zusätzlich auch aus den beiden Voreinstellungen Mittweida und Rosswein gewählt werden. Der Bereich Details enthält jedoch Parameter speziell die Suche betreffend. So gibt es die Option, nur Inserate eines bestimmten Nutzers zu finden, oder die Einträge (in Kombination mit dem persönlichen Profil) nach dem Rauchverhalten oder dem Ge-

schlecht der Inserenten zu filtern. Die letzten beiden Angaben können im persönlichen Profil schon festgelegt werden, und werden dann hier automatisch übernommen (das Auswahlfeld müsste bei Bedarf aber erst aktiviert werden).

Alle verfügbaren Unterscheidungsparameter können auf Wunsch auch unberücksichtigt bleiben bzw. passiv geschaltet werden, größtenteils indem die entsprechende Checkbox vor dem Formularelement gesetzt wird. Eine Ausnahme bilden dabei die Auswahlfelder betreffend Toleranz von Datum und Uhrzeit, sowie Raucherpräferenz und Geschlecht der Mitfahrer. Diese sind aufgrund ihrer doch eher sekundären Bedeutung von vornherein als inaktiv definiert, und müssten erst bei Bedarf durch die Checkbox aktiviert werden. Durch die entsprechende Kennzeichnung sind die jeweils verfügbaren Optionen hinsichtlich ihrer Funktion aber klar erkennbar, und sorgen keinesfalls für Verwirrung. Werden die Textfelder, sowie die Auswahl der Art bei der Suche ignoriert (d.h. wird keine Angabe getätigt, sowie das „egal“ nicht gesetzt), sind die betreffenden Parameter bei der Bearbeitung der Anfrage automatisch als „egal“ definiert (Hinweis wird entsprechend auf der Ergebnisseite eingeblendet).

- **Personensuche**

Ab und zu möchte man beispielsweise in Erfahrung bringen, welche Fahrten ein bestimmter Nutzer gerade anbietet, oder man möchte mehr über eine bestimmte am Dienst beteiligte Person wissen. In diesen Fällen bietet es sich an, direkt nach der Person selbst suchen zu können, anstatt erst über die Inseratssuche gehen zu müssen.

Diese Möglichkeit steht den Nutzern direkt auf der Startseite des Dienstes zur Verfügung. Name und/oder Vorname oder Seminargruppe können als Suchkriterien genutzt werden.

- **Suchergebnisse**

Die Resultate werden zunächst allgemein aufgelistet. Die konkreten Einzelheiten der jeweiligen Einträge können separat über eine Detailseite abgerufen werden. Außerdem ist es möglich, die Ergebnisliste nach verschiedenen Kriterien, das heißt den einzelnen Spalten der jeweiligen Tabelle, zu sortieren.

2.3.4 Buchen

Hat ein Nutzer eine passende Fahrt gefunden, die er gerne in Anspruch nehmen würde, dann sollte das betreffende Inserat gebucht werden. Damit wird die benötigte Anzahl an Plätzen fest reserviert, und man hat nach der Fahrt die Möglichkeit, eine Bewertung vorzunehmen.

Auf der Detailseite des jeweiligen Inserats ist im Abschnitt „Zusatzoptionen“ der Link zum zugehörigen Buchungsformular zu finden. Wurde das Inserat jedoch schon gebucht, erscheint ein Hinweis, die Buchung mit Hilfe der Verwaltungsübersicht falls gewünscht zu ändern.

Abhängig von Art und Frequenz des Inserats bietet das Formular verschiedene Optionen, die Buchung zu beeinflussen.

- **Gesuche**

Gesuche müssen nach ihrer Definition, das heißt im „Ist-Zustand“ gebucht werden. In diesem speziellen Fall existieren konkrete Parameter, die mindestens erfüllt sein müssen, um die Fahrt zu ermöglichen. Hierbei ist die Buchung vor allem als Voraussetzung für die spätere Bewertung zu sehen, aber auch um alle Fahrten mit eigener Beteiligung in der Verwaltung managen zu können.

- **Angebote**

Bei Angeboten gibt es jedoch bestimmte Auswahlmöglichkeiten. Dabei müssen folgende Szenarien unterschieden werden:

- Einmaliges Angebot
- Regelmäßiges Angebot auf einen Wochentag bezogen (montags dienstags, etc.)
- Regelmäßiges Angebot auf mehrere Tage bezogen (täglich, werktags)

Bei **einmaligen Angeboten** hat der Nutzer die Chance, Start- und Zielort entsprechend der verfügbaren Orte zu frei zu wählen (je nach dem, ob Zwischenstationen eingetragen sind). Jedoch dürfen Start und Ziel nicht identisch sein. Deshalb passt sich die Liste verfügbarer Zielorte immer der Auswahl des Starts an. Außerdem kann bei einmaligen Inseraten noch die Menge der benötigten Plätze gewählt werden, abhängig von der noch verfügbaren Anzahl.

Alle **regelmäßigen Angebote** beinhalten zusätzlich (zu Orts- und Platzwahl) noch die Komponente der Frequenzdifferenzierung. Bei einem festen Wochentag kann das Angebot regelmäßig, oder auch als Einzelfahrt gebucht werden. Bezieht sich das Angebot auf mehrere Tage (betrifft täglich und werktags), gibt es darüber hinaus noch die Möglichkeit einzelne Tage daraus regelmäßig zu buchen. Bei Einzelfahrten muss das gewünschte Datum der Fahrt angegeben werden; danach wird überprüft, wie viele Plätze an diesem Tag noch frei sind. Sollen ein oder mehrere Tage regelmäßig in Anspruch genommen werden, kann der Nutzer diese über Checkboxen auswählen. Danach wird berechnet wie viele Plätze mindestens für den gesamten Zeitraum bereitgestellt werden könnten. So kann es passieren, dass an einem Teil der Tage mehr Plätze frei sind, als im Endeffekt angezeigt werden (oder das laut Anzeige keine Plätze frei sind, aber in Wirklich-

keit zum Teil noch etwas offen ist). Jedoch muss die Platzanzahl an jedem der gewünschten Tage verfügbar sein, weshalb die niedrigste Zahl zur Berechnung hergenommen wird. Die uneinheitlichen Mengen können sich aus eben solchen Teilbuchungen ergeben.

- **Verwaltung**

Alle Buchungen - sowohl die eigenen, als auch die bei einem selbst vorgenommenen – werden mittels einer Verwaltungsübersicht gemanagt. Wie bei den Inseraten selbst können Änderungen vorgenommen, Details betrachtet und Einträge gelöscht werden. Dabei ist es hier möglich, die Inserats- und die Buchungsdetails direkt aus der Übersicht abzurufen. Buchungsänderungen werden analog zu den Inseraten über das ursprüngliche Formular geregelt, indem die entsprechenden Daten eingelesen werden. Es kann ja nun durchaus vorkommen, dass für ein Angebot mehrere Buchungen vorliegen. Zur besseren Übersicht ist es deshalb möglich, über die Inseratsdetails alle Buchungen für das betreffende Inserat auf einer separaten Seite abzurufen. Bei allen Fremdbuchungen können erwartungsgemäß nur die jeweiligen Details betrachtet werden. Die Buchungen selbst sind in diesem Fall nicht beeinflussbar.

- **Buchungsdetails**

Diese Seite dient zur Ansicht bzw. Kontrolle der konkreten Details einer Buchung. Neben den Einzelheiten zur Fahrt selbst, ist es so auch möglich die betreffende Identität zu erfahren. Hat der Nutzer selbst eine Buchung vorgenommen, wird als Identität der Name des Inserenten angegeben. Wurde ein Angebot oder Gesuch bei dem Nutzer gebucht, wird als Identität der Name des Buchenden angegeben. Über den entsprechenden Link gelangt der Nutzer zur Profilseite der Person.

Folgende Informationen zum Buchungsvorgang werden dargestellt:

- Identität (Inserent oder Buchender)
- Frequenz
- Datum
- Strecke (Startort und Zielort)
- Platzanzahl
- Buchungs- bzw. Änderungsdatum

- **Buchung löschen**

Es können immer nur die Buchungen gelöscht werden, die der Nutzer selbst vorgenommen hat. Im Zuge dessen werden die zuvor gebuchten Plätze wieder freigegeben, und der temporäre Bewertungseintrag wird gelöscht

2.3.5 Bewerten

In den meisten Fällen kennt der interessierte Nutzer die Person nicht, welche das Inserat erstellt hat. Ein Bewertungssystem bietet so die Möglichkeit mehr über den betreffenden Nutzer zu erfahren. Basierend auf den Erfahrungsberichten bisheriger Fahrer und Mitfahrer existiert so ein Informationssystem, welches im Laufe der Zeit sowohl an Umfang, als auch zunehmend an Aussagekraft gewinnt durch eine wachsende Anzahl an unabhängigen Bewertungen. Dabei muss aber unbedingt darauf geachtet werden, die Nutzung unkompliziert zu gestalten, damit diese Funktion überhaupt angenommen wird (denn ohne Beteiligung verliert die Funktion an Bedeutung).

▪ Grundlage

Während der Buchung eines Inserats wird automatisch ein temporärer Bewertungseintrag erstellt. Somit ist der Missbrauch, das heißt die Bewertung durch unbefugte Dritte, ausgeschlossen. Die vorhandene Nutzer-Nutzer Zuordnung wird herangezogen, um die jeweilige Bewertung nur für die an der konkreten Buchung beteiligten Personen zu ermöglichen.

Einmalige Fahrten können auch nur einmal bewertet werden. Nach der erfolgreichen Bewertung wird der temporäre Eintrag gelöscht, und es bleibt nur der permanente Datensatz übrig. Damit ist eine erneute Beurteilung nicht möglich.

Regelmäßige Fahrten können jedoch mehrfach bewertet werden, da sich die Meinung zur betreffenden Person im Laufe der Zeit durchaus ändern kann. Dabei ist pro Woche maximal eine Bewertung möglich. Das heißt, nimmt der Nutzer eine Beurteilung vor, wird der nächstmögliche Bewertungszeitpunkt errechnet (aktueller Timestamp + eine Woche) und dieser wird dann in die Datenbank eingetragen. Alle Bewertungen können frühestens ab dem ersten Tag nach der Fahrt vorgenommen werden. Das heißt, Einzelfahrten können ab dem ersten Tag nach dem festgelegten Datum bewertet werden. Für alle regelmäßigen Fahrten wird generell ein Zwischenraum von einer Woche ab eingetragendem Datum (wenn bei der Buchung noch in der Zukunft liegend) oder dem Buchungsdatum definiert. Damit soll sichergestellt werden, dass die Fahrt vor der ersten Bewertung mindestens einmal stattgefunden hat.

▪ Verwaltung

In dieser Übersicht findet man sowohl die bereits abgeschlossenen, als auch die noch abzugebenden Bewertungen. Erhaltene und selbst abgegebene Bewertungen können im Detail betrachtet werden, wobei auch bei diesen Auflistungen eine Sortierung möglich ist. Offene Bewertungen sind wegen der besseren Übersicht in drei Kategorien eingeordnet:

- Freigegebene Bewertungen
- Noch nicht freigegebene Bewertungen
- Derzeit gesperrte Bewertungen

Nach der Buchung sind alle Einträge erst einmal als „noch nicht freigegeben“ definiert. Das heißt, die Fahrt hat noch nicht stattgefunden, und kann deshalb auch noch nicht bewertet werden. Jedoch hat man so einen Überblick, welche Fahrten später noch zur Bewertung anstehen. Ist die Fahrt vorüber, wird die Bewertung

ab dem nächsten Tag als „freigegeben“ eingestuft. Regelmäßige Fahrten, die bereits vor kurzem schon einmal bewertet wurden, sind ab dem Zeitpunkt der Bewertung für eine Woche gesperrt und dementsprechend aufgelistet.

- **Formular**

Die Bewertung selbst ist recht simpel. Im oberen Teil des Formulars sind die Eckdaten der Fahrt nochmals aufgeführt, und darunter ist der Bewertungsabschnitt zu finden. Der Nutzer hat die Möglichkeit, je nach Art des zu Grunde liegenden Inserats seinem Fahrer bzw. Mitfahrer für die Kriterien „Zuverlässigkeit“ und „Auftreten“ eine Benotung von 1 (schlecht) bis 5 (sehr gut) zu geben. Zusätzlich kann auch noch ein kurzer einzeliger Kommentar verfasst werden, um die Entscheidung näher zu beschreiben.

2.3.6 Detailinformationen zu den Nutzern des Dienstes

Abseits der eigentlichen Daten zu Fahrt selbst, werden noch zusätzliche Informationen benötigt, um den Dienst sinnvoll nutzen zu können. Grundlegend wären dies der Name des Nutzers und dessen E-Mail Adresse. Aber weitere Daten - welche den Nutzer oder dessen Fahrzeug an sich näher beschreiben - tragen zu besseren Vermittlungschancen bei, lassen einen unbekannten Nutzer vertrauter wirken, oder vereinfachen beispielsweise das Auffinden eines unbekannten Fahrzeugs am Treffpunkt. Diese Angaben bei jedem einzelnen Inserat wiederholen zu müssen, wäre auf Dauer mühselig. Außerdem könnte ein unabhängiges Profil auch separat von sämtlichen Angeboten oder Gesuchen betrachtet, oder auch im Zuge von Buchungs- und Bewertungsdetails verlinkt werden.

Deshalb werden bei diesem Mitfahrservice zwei eigenständige Profile bereitgestellt, eins für die personenbezogenen Informationen und ein weiteres zur Beschreibung des Fahrzeugs. Was die persönlichen Daten betrifft, gibt es ein paar schon erwähnte Pflichtangaben, ohne deren Komplettierung der Dienst nur oberflächlich genutzt werden kann (beispielsweise keine Buchungen oder Bewertungen möglich). Das Fahrzeugprofil beinhaltet jedoch gänzlich freiwillige Kenngrößen. Die vorgenommenen Angaben werden schließlich auf einer Profilseite zusammengefasst. Diese Seite kann dann je nach Bedarf inseratsunabhängig abgerufen oder implementiert werden.

- **Persönliches Profil**

Dieses Formular ist in folgende Abschnitte unterteilt:

- Kontaktinfos
- Persönliche Einstellungen
- Fahrteinstellungen

Für die Funktionalität des Dienstes an sich sind die **Kontaktinformationen** am wichtigsten. Pflichtangaben wie Name und E-Mail Adresse sind hier vertreten, um die grundlegende Kommunikation zwischen den Nutzern und für die Benachrichtigungsmails sicherzustellen. Alle Angaben müssen jedoch auf eigene Verantwortung getroffen werden, denn in dieser Implementierung wird keine Überprüfung der Namen an sich durchgeführt (es werden nur dienstinterne Datentabellen genutzt). Um den Dienst auf Dauer für alle Interessenten nutzbar zu belassen, sollten alle Einträge nur mit lauterer Absichten vorgenommen werden (was aber vom Niveau eines Studenten vorausgesetzt werden sollte!). Der Nutzerlogin wird aber bei allen Einträgen gespeichert, und könnte im Notfall herangezogen werden.

Es können aber auch relevante Telefonnummern eingetragen werden, um einen direkteren Kontakt zu ermöglichen. Die Seminargruppe ist nützlich, um die Personensuche auch auf diesen Parameter anwenden zu können (welche Mitglieder meiner Seminargruppe beteiligen sich am Mitfahrdienst?). Da aber nicht jeder Nutzer seine Kontaktdaten für alle Personen einfach so zugänglich machen will, gibt es die Option, diese Daten nicht zu veröffentlichen. Das heißt, nur Personen die bei ihnen oder bei denen sie selbst ein Inserat gebucht haben können die angegebenen Kontaktinformationen frei einsehen. Alle Interessenten, welche sich gerade auf der Suche nach einem passenden Angebot oder Gesuch befinden, können jedoch immer mindestens das Kontaktformular nutzen (auch wenn die relevanten Angaben nicht verfügbar sind).

Die **persönlichen Einstellungen** befassen sich mit freiwilligen Angaben zur eigenen Person, die für die spezialisierte Suche oder generell als Auswahlkriterien im Dienst relevant sein könnten. Ist der Nutzer ein Raucher? Ist der Nutzer männlich oder weiblich? Auch die Auskunft über die Sprachen in denen der Nutzer kommunizieren kann, erweist sich in einem so internationalen Nutzungsumfeld wie der Hochschule durchaus als hilfreich.

Die restlichen **Einstellungen** haben den Zweck, die persönlichen Präferenzen die **Fahrt** an sich betreffend festzulegen. Hat der Nutzer keine bestimmten Bevorzugungen, kann auch die Option „egal“ gewählt werden. Diese Angaben sind dann auf der Profilseite wieder zu finden. Die Einstellungen zum Thema Raucher und Geschlecht werden automatisch für die Detailsuche übernommen, und müssen so nicht jedes Mal geändert werden. Achtung: vor allem bei einer geringen Anzahl an Inseraten sollte man diese Angaben variieren und nicht unbedingt darauf bestehen; es kann ja auch sein, dass nicht alle Personen die entsprechenden Angaben vervollständigt haben!

- **Fahrzeugprofil**

Dieses Formular besteht nur aus dem Abschnitt Fahrzeugdetails. Zu einem Fahrzeug reicht eine begrenzte Zahl an zweckmäßigen Beschreibungsparametern; aber ein eigenes Profil ist trotzdem sinnvoll, um die Inhalte thematisch logisch trennen zu können.

Das Auto kann mit folgenden Parametern näher beschrieben werden: Kennzeichen, Automarke, Autotyp, Autofarbe und freie Plätze. Bei den Plätzen wird dies als Standardangabe verstanden, die bei der Erstellung eines Inserats automatisch übernommen wird (kann natürlich im Inserat wieder geändert werden). Außerdem gibt es in diesem Formular noch ein Kommentarfeld, um das Fahrzeug noch mit ein paar zusätzlichen Worten näher beschreiben zu können.

Genau wie bei dem persönlichen Profil die Kontaktdaten betreffend, kann auch hier festgelegt werden, ob diese doch eher sensiblen Daten für alle Personen freisichtbar sein dürfen, oder ob dies nur für Personen gilt, die an Buchungen mit einem selbst beteiligt sind.

- **Profilseite**

Die Details aus den beiden Profilformularen werden hier zentral versammelt aufgeführt und können von allen Personen eingesehen werden. Einschränkungen gibt es in dem Fall, dass die sensiblen Daten als „nicht öffentlich“ gekennzeichnet wurden (entsprechende Hinweise werden eingeblendet). Der grundlegende Kontakt zum Nutzer ist aber in jedem Fall durch das implementierte Mailformular sichergestellt. Weiterhin gibt es über jedes Profil, unabhängig der getätigten Angaben, die Möglichkeit, alle bisherigen Bewertungen im Detail zu betrachten (vorausgesetzt es liegen bereits Bewertungen vor!). Darüber hinaus kann hier direkt auf alle vom betreffenden Nutzer erstellten Angebote und Gesuche zugegriffen werden, ohne erst über die Suchfunktion gehen zu müssen.

Diese Profilseiten an sich sind entweder über die Personensuche erreichbar, oder durch die Implementierung in den Inseratsdetails, den Buchungsdetails und den Bewertungsdetails auch dort zugänglich. Der Nutzer selbst kann seine eigene Profilsansicht zusätzlich direkt über einen Link auf der Startseite aufrufen.

2.3.7 Kontaktaufnahme/Benachrichtigung

Ein weiterer Eckpunkt dieses Mitfahrdienstes ist die Möglichkeit der Kommunikation. Interessierte Nutzer müssen sich mit dem Inserenten ihres ausgewählten Angebotes oder Gesuches in Verbindung setzen können, etwa um eventuelle Fragen vor der Buchung klären zu können. Aber auch der Inserent selbst muss die Personen, die bei ihm gebucht haben, beispielsweise im Fall einer anstehenden Änderung erreichen können. In diesem Zusammenhang sind bei dem hier vorliegenden Dienst drei entsprechende Komponenten vorgesehen.

- **Kontaktdaten**

Sind die Kontaktdaten freigegeben, oder ist der Nutzer an einer Buchung beteiligt, steht mindestens die E-Mail Adresse (= Pflichteingabe) zur Verfügung. Das Profil bietet aber auch Platz für die Eintragung von Mobilfunk- und/oder Festnetznummer, um aktiv mit anderen Personen in Kontakt treten zu können.

- **Kontaktformular**

Diese Kontaktmöglichkeit ist zu jedem Zeitpunkt nutzbar. Über die Profilseite eines Nutzers erhält man Zugang zum Formular über den Link „Nachrichten schicken“. In der hier realisierten Form stellt sich diese Funktionalität als recht simpel dar. Ein schlichtes Textfeld genügt, um eine Mitteilung weiterzuschicken. Die eigene im Profil angegebene Mailadresse wird herangezogen, und zusammen mit der Empfängeradresse und dem Text aus dem Formular zu einer E-Mail zusammengefügt und verschickt.

- **Benachrichtigungsmails**

Ein Mindestmaß an Kommunikation wird durch die automatische Versendung von E-Mails zur Benachrichtigung verschiedener Ereignisse im Dienstablauf gewährleistet. Diese Funktion besitzt durchaus Potential für Erweiterungen, aber bei dem in dieser Arbeit festgehaltenen Zustand sind für folgende Ereignisse Benachrichtigungen implementiert worden:

- Inserat wurde gebucht
- Inserat wurde geändert
- Inserat soll geändert werden – Änderungen sind bei vorliegenden Buchungen nicht ohne weiteres möglich
- Inserat wurde gelöscht
- Buchung wurde geändert
- Buchung wurde gelöscht
- Bewertung wurde abgegeben

Werden Buchungen vorgenommen, geändert und gelöscht oder wird eine Bewertung abgegeben, schickt die korrespondierende Funktion sofort eine Benachrichtigung vom auslösenden Nutzer zum Inserenten (1-1 Beziehung). Wird jedoch ein bestehendes Inserat modifiziert oder gelöscht, muss erst überprüft werden, ob entsprechende Buchungen vorliegen. Ist dies der Fall, wird eine Mitteilung an alle betreffenden Nutzer geschickt (1-n Beziehung).

Durch diese Mitteilungen werden die beteiligten Nutzer immer über wichtige Ereignisse oder Änderungen auf dem Laufenden gehalten. In Verbindung mit der Buchungsfunktion ist es so beispielsweise nicht mehr zwingend notwendig, vor der Fahrt mit dem Inserenten in Verbindung treten zu müssen (wenn die Informationen aus den Inseratsdetails ausreichend sind). Der Inserent ist über das Interesse anderer Nutzer informiert, und kann bei Bedarf auch selbst Initiative ergreifen und die Buchenden kontaktieren.

3 Programmierungsumgebung

3.1 Übersichtsdiagramm Programmseiten / Bibliotheken / Klassen

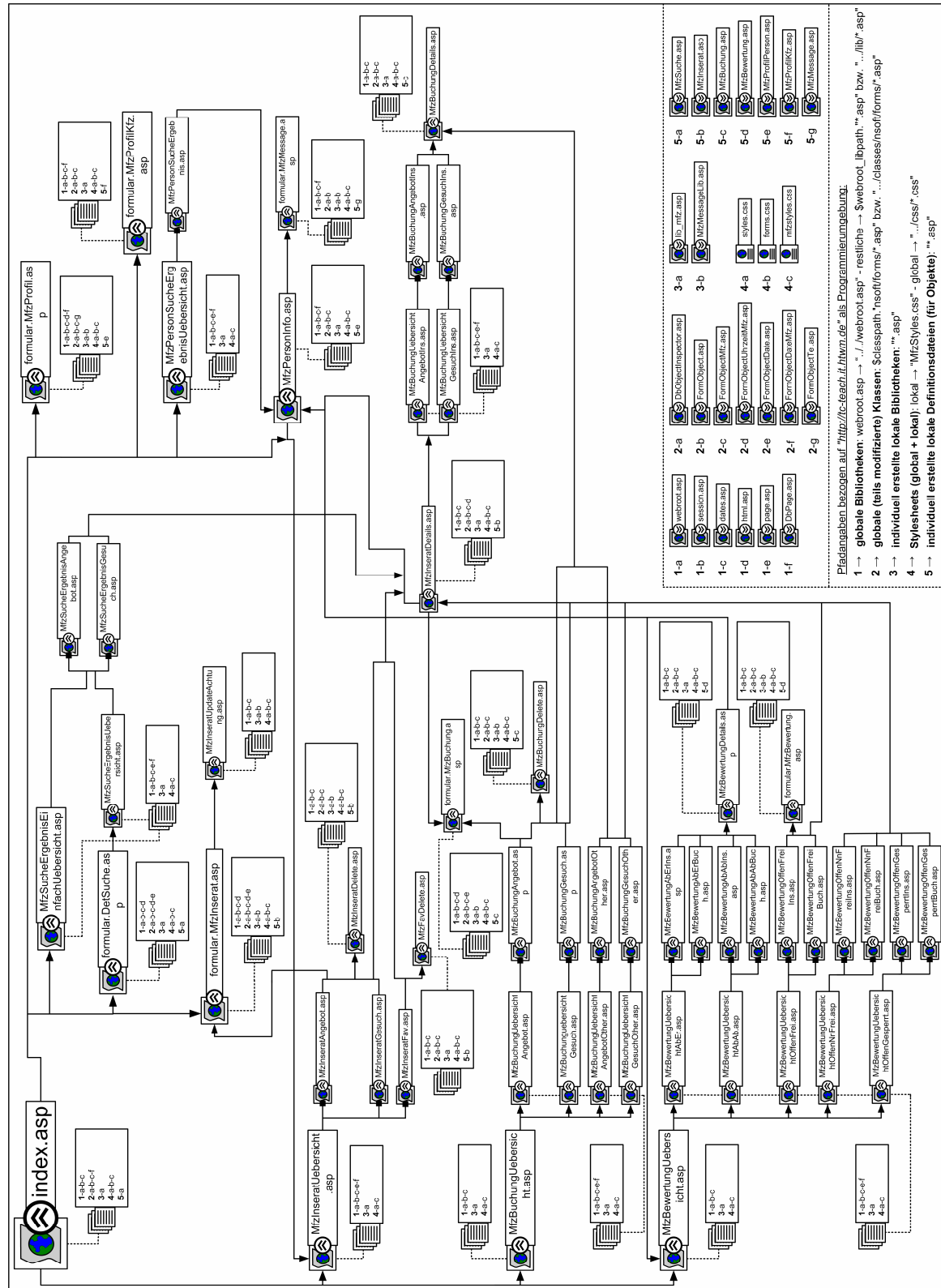


Abbildung 3-1 Übersicht Programmseiten / Bibliotheken / Klassen (Microsoft Visio 2007)

3.2 Verwendete Bibliotheken / Klassen

Aufgelistet werden hier nur die Dateien, welche separat eingebunden sind. Unter Umständen können in den einzelnen Bibliotheken und Klassen weitere Implementierungen/Abhängigkeiten vorliegen. Die hier getätigten Angaben beziehen sich auf *tc-teach* als Programmierumgebung.

- **globale Bibliotheken für die Programmierung innerhalb der Hochschulwebsite**

- (1) *webroot.asp*

- Die zentralen Pfadeinstellungen für die korrekte Einbindung und Funktionalität der zugehörigen Bibliotheken, Klassen und Dateien befinden sich in dieser Bibliothek.

- (2) *session.asp*

- Ähnlich *webroot.asp* wird diese Datei für die einwandfreie Funktionalität des Dienstes benötigt. In der hier verwendeten Programmierumgebung (*tc-teach*) ergaben sich ohne Einsatz dieser Bibliothek etwa Fehler bei Datenbankabfragen oder der Verarbeitung bestimmter Parameter.

- (3) *dates.asp*

- In dieser Bibliothek befinden sich diverse Funktionen für den Umgang mit Zeitparametern. Die Datei *datetime.asp* stellt dabei zwar eine aktuellere Version dar, jedoch hat sich gezeigt, dass ohne die Einbindung von *dates.asp* Fehler in der Verarbeitung und Darstellung der einzelnen Seiten auftreten. Wahrscheinlich existieren im Ablauf noch entsprechende Abhängigkeiten in verwendeten Bibliotheken oder Klassen.

- (4) *html.asp*

- Verschiedene HTML - Funktionen sind in dieser Datei enthalten. Bei dem hier entwickelten Dienst wird bei der Programmierung der Formulare darauf zurückgegriffen (siehe Auswahlfelder). In diesem Fall erfolgt die konkrete Anwendung in den Klassendateien *FormObject.asp* und *FormObjectMfz.asp*.

- (5) *page.asp*

- Ähnlich zu der vorherigen Bibliothek (*html.asp*) werden auch in dieser Datei einige HTML - und Anzeige - Funktionen zusammengefasst, welche die Darstellung einer Seite betreffen bzw. beeinflussen können. Bei diesem Dienst existieren Abhängigkeiten bei der Programmierung jeglicher Seiten, welche eine Auflistungstabelle enthalten (wie etwa bei den einzelnen Übersichtsseiten). So wird dort die Funktion `switchRowColor()` eingesetzt, um eine besser lesbare Anzeige der Tabellen zu ermöglichen, und zwar durch Farbwechsel des Hintergrundes jeder neuen Zeile.

- (6) *DbPage.asp*

- Auch diese Bibliothek befasst sich mit diversen Routinen zur Manipulation des Erscheinungsbildes einer Seite. Jedoch bezieht sich dies hier spezieller auf Funktionen die objektbasierte Programmierung betreffend. Die entspre-

chende Handhabung von Daten bzw. Parametern wird behandelt, wie etwa deren Aufrechterhaltung bei möglichen Selbstaufrufen der Seite (und der damit verbundenen Sicherung einer jeweils gültigen Datenbankabfrage). Konkret für den Mitfahrdienst werden beispielsweise die Funktionen `DbPage()` und `CheckBarCss()` verwendet (siehe Übersichts-/Ergebnistabellen oder Seiten mit passiv schaltbaren Abschnitten). Letztere Funktion realisiert dabei ein optisches Element zum Ein- und Ausblenden der jeweiligen Tabellen oder Abschnitte.

- **globale Klassen zur Realisierung der Formularseiten (Standardvarianten)**

- (1) **FormObject.asp**

Hierbei handelt es sich um eine der beiden zentralen Dateien, welche die grundlegende Realisierung und Verarbeitung von objektbasierten Formularseiten ermöglichen. Die Funktion `RenderControl($obj)` ist dabei wohl als eine Art Kernstück zu betrachten, da über diese Standardvariante alle für das Formular definierten Objekte in konkreten Quelltext umgesetzt und damit in ihrer Funktion realisiert werden können. Die entsprechenden Hinweistexte und Elementbezeichnungen werden dann mit den Funktionen `RenderInfo()` und `RenderLabel()` eingepasst.

- (2) **FormObjectTel.asp**

Die Klasse *FormObjectTel* ist eine spezielle Erweiterung der Klasse *FormObject*. Hierbei handelt es sich um eine spezielle Vorlage für Formularobjekte, welche zur Eingabe einer Telefonnummer vorgesehen sind. Die wichtigsten Details sind dabei der Typ des Elements (hier: *input*) und das Gültigkeitsmuster (über den Parameter *valid* gesteuert). Soll ein Objekt nach diesen Vorgaben erstellt werden, muss bei dessen Definition darauf geachtet werden, anstatt wie normalerweise `$this->AddProp` die Zuweisung `$this->AddPropClass` mit entsprechender Angabe der speziellen Klasse zu verwenden.

- (3) **FormObjectDate.asp**

Auch hierbei handelt es sich um eine Formatvorlage eines Formularobjektes, das einem bestimmten Zweck dienen soll. In diesem Fall ist das die Eingabe bzw. Auswahl eines Datums. Grundsätzlich geschieht dies über eine normale Textzeile. Jedoch soll dem Nutzer auch die Möglichkeit geboten werden, zusätzlich zur normalen Texteingabe, ein Datum direkt über eine Art Kalender zu wählen (wie es beispielsweise auch bei *raumobil.de* in der erweiterten Suche implementiert ist). Deswegen beinhaltet diese Klassendatei unter anderem auch eine angepasste Variante der Funktion `RenderControl($obj)`, um nur in diesem speziellen Fall das Kalendersymbol einzublenden.

- (4) **DbObjectInspector.asp**

Für die Umsetzung objektbasierter (Formular-)Seiten stellt dies die zweite wichtige Datei dar. Der wesentliche Teil sind die beiden Funktionen `AddProp()` und `AddPropClass()`, welche die Objektdefinitionen selbst realisieren. Dies ist prinzipiell zwar auch möglich über `new FormObject(...)` und

der separaten Angabe jedes einzelnen Parameters, aber durch die beiden genannten Funktionen lässt sich das Ganze auf eine Zeile reduzieren.

- **individuell erstellte Bibliotheken**

- (1) `lib_mfz.asp`

In dieser Datei befinden sich Hilfsfunktionen, die speziell für die Nutzung innerhalb der Programmierung des Mitfahrdienstes erstellt wurden. Die ersten fünf Funktionen basieren auf PHP, der Rest auf JavaScript.

- `wochentag($time)` → In dieser vielfach innerhalb des Mitfahrdienstes zur Anwendung gebrachten Funktion wird vom übergebenen Zeitpunkt (in Form eines Timestamps) der entsprechende Wochentag ermittelt. Als Ergebnis wird jeweils die Kurzform ausgegeben, wie etwa „Mo“ für Montag, „Di“ für Dienstag usw.
- `wochentag_lang()` → Hierbei handelt es sich um eine von zwei Funktionen, die aus der Bibliothek der vorherigen Version des Mitfahrdienstes übernommen wurden (so musste diese Datei nicht separat eingebunden werden). Der aktuelle Wochentag wird an dieser Stelle in ausgeschriebener Form zurückgeliefert. Diese Funktion kommt nur bei der auf jeder Seite des Dienstes verwendeten Anzeige des aktuellen Zeitpunktes (Datum + Uhrzeit) zum Einsatz (diese Anzeige wurde aus dem alten Mitfahrdienst übernommen).
- `monat_lang()` → Analog zur vorherigen Funktion stammt diese ebenfalls aus der Version des Mitfahrdienstes von Sandro Gerstenberger. Jedoch wurde in diesem Fall eine kleine Änderung vorgenommen, indem hier genau wie bei `wochentag_lang()` durchgängig die Funktion `date()` zu Hilfe genommen wird. So wird bei diesen beiden sinnverwandten Funktionen nebenbei auch eine strukturelle Angleichung erreicht. Wie der Name schon andeutet wird in diesem Fall anstatt dem jeweilig aktuellen Wochentag der Monat ausgegeben (auch hier zur Verwendung im Seitenkopf zur Zeitanzeige).
- `browserweiche()` → Über diese Funktion wird grundlegend der bei der Nutzung des Studentenportals verwendete Browsertyp ermittelt. In der vorliegenden Umsetzung kommt es primär darauf an, für einen speziellen Fall der Anwendung zu ermitteln, ob es sich bei dem verwendeten Browser um *Firefox* handelt oder nicht. Da aber auch auf diverse Browsertypen explizit geprüft wird, kann diese Funktion durch Ergänzung eines zweiten Ausgabeparameters auch anderweitig als Weiche genutzt werden (in diesem Projekt bspw. bei Fehlermeldungen/Mitteilungen am oberen Bildschirmrand benötigt – aufgrund abweichender bzw. leicht verschobener Darstellung im *Internet Explorer*).

-
- `sort_icon($richtung)` → Bei der Darstellung der Übersichts- bzw. Ergebnistabellen soll nach erstmaliger manueller Sortierung die Richtung (aufsteigend/absteigend) durch die Einblendung einer entsprechenden Grafik verdeutlicht werden - außerdem ist so schneller ersichtlich, nach welchem Parameter als letztes sortiert wurde. Die Funktion bindet hierbei je nach gerade übergebener Richtung der Abfrage die entsprechende Grafik im Tabellenkopf ein.
 - `disableenable (de)` → Mit dieser Funktion kann ein bestimmtes Element im Wechsel aktiviert und deaktiviert werden (beispielsweise in Kombination mit einer Checkbox, jedes Mal wenn diese angeklickt wird).
 - `disableenable1 (de)` → Das Prinzip ist dasselbe wie bei der vorherigen Funktion, nur dass diese hier für die Deaktivierung einer Gruppe Radiobuttons mit zwei Auswahloptionen vorgesehen ist (siehe Parameter „Art“ bei der Detailsuche).
 - `disableenable2 (de, de1)` → Diese Funktion ist eine Kombination der Beiden vorherigen. Das heißt hiermit werden gleichzeitig ein einfaches Element, sowie eine Zweiergruppe Radiobuttons deaktiviert bzw. aktiviert.
 - `disableenable3 (de, de1, de3)` → Diese spezielle Anpassung kommt im detaillierten Suchformular zum Einsatz. Dort können Datum und Uhrzeit inklusive Toleranzwert gewählt werden. Bei der (De)Aktivierung des kompletten Kriteriums muss aufgrund der verwendeten Checkbox (Aktivierung der Toleranzwerte) deren Zustand zwecks korrekter Wiederherstellung gesondert betrachtet werden.
 - `disable_stop (de, de1, de2)` → Da zur Eintragung bzw. Modifikation von Angeboten und Gesuchen das gleiche Formular genutzt wird, jedoch bei Gesuchen keine Zwischenhaltepunkte angegeben werden, müssen die betreffenden Elemente gegebenenfalls deaktiviert werden. Diese Funktion schaltet alle drei *input*- Elemente aus, sobald der Radiobutton für die Option „Gesuch“ als Art des Inserats angeklickt wird.
 - `enable_stop (de, de1, de2)` → Mittels dieser Funktion werden bei der Bearbeitung eines Inserats unter Auswahl der Art „Angebot“ alle Eingabefelder für Zwischenhaltepunkte aktiv geschaltet.
 - `setinputort (ie, si)` → Diese Funktion realisiert im Grunde eine simple Wertzuweisung. Zur besseren Orientierung wurde jedoch der Name entsprechend des spezifischen Einsatzortes gewählt. Im Formular für die detaillierte Suche besteht die Möglichkeit, bei Start- und Zielort mittels Radiobutton aus den voreingestellten Werten „Mittweida“ und „Rosswein“ zu wählen. Dabei wird die getätigte Auswahl auf die zugehöri-

ge Textzeile übertragen. Der Name der Textzeile und der Wert der Auswahl werden übergeben.

- `setcolor (se, sv)` → Hierbei wird der Farbwert eines Elements als Folge einer bestimmten Aktion geändert. Die Eingabefelder der einfachen Suchfunktion auf der Startseite sind mit einem grauen Definitionstext vorbelegt. Wird eine dieser Textzeilen zur konkreten Suche genutzt, erscheint der Eingabetext mit Hilfe dieser Funktion wieder in der standardmäßig schwarzen Farbe. Als Funktionsparameter müssen hierbei das betroffene Element, sowie der gewünschte Farbwert übergeben werden.
- `radiouncheck (re)` → Wurde bei der Detailsuche mittels Radiobutton ein voreingestellter Start-/Zielort gewählt, erhält die zugehörige Textzeile den passenden Wert. Möchte der Nutzer aber lieber doch einen anderen Ort eintragen (per Handeingabe), so wird mittels dieser Funktion beim Klick in die Textzeile die Markierung des Radiobutton entfernt. Bei der Suche zählt letztendlich der Inhalt der Textzeile, aber um Verwirrung zu vermeiden bzw. um ein klares Erscheinungsbild zu wahren, sollte der Radiobutton nur aktiv sein, wenn eine eindeutige Zuordnung gewährleistet ist.
- `changecheckbox (cb1)` → Um während der laufenden Bearbeitung eines Formulars durchgehend korrekt mit Checkboxes arbeiten zu können, wird momentan diese Hilfsfunktion an einigen Stellen benötigt (d.h., wenn eine Speicherung oder ein *submit* des Formulars vor dem Abschluss der Bearbeitung vorgesehen ist). Bei Aktivierung eines solchen Elements und einem darauffolgenden *submit* – also bei einem Wechsel von 0 auf 1 – wird die Änderung noch problemlos registriert. Soll danach die Checkbox aber wieder deaktiviert werden – also von 1 auf 0 wechseln – kommt ohne den Einsatz dieser Funktion keine Änderung zustande. Eine Ausnahme ergibt sich zwar beispielsweise beim Aufruf eines bestehenden Inserats, wobei dort eine im aktiven Zustand geladene Checkbox ohne Verwendung dieser Funktion einmalig deaktiviert werden könnte. Jedoch funktioniert das beim zweiten Versuch schon nicht mehr (→ keine durchgehende Funktionalität garantiert!).
- `setAnker (ae)` → Besitzen die Übersichts- bzw. Suchergebnisseiten mehr als eine Tabelle - siehe Trennung nach Angeboten/Gesuchen bei der Inseratssuche - dann würde bei einem Sortierungsvorgang (nach dem *reload*) normalerweise die Ansicht zum Seitenanfang zurückkehren - auch wenn eine untere Tabelle sortiert wird. Diese würde sogar ganz aus dem Fokus verschwinden, bei einer entsprechend hohen Menge an Einträgen in der oberen Tabelle. Um dies zu verhindern, werden so genannte „Anker“ verwendet, wobei jede Kopfzeile einer

sortierbaren Spalte mit einem entsprechenden Wert belegt ist (bei der jeweils oberen Tabelle wurde darauf verzichtet). Jedoch kann in diesem Fall nicht mit dem in HTML definierten Anker-Element gearbeitet werden, weil damit nach Aktivierung eines bestimmten Verweises (hier bei der manuellen Sortierung) der betroffene Parameter „blockiert“ wäre. Das heißt, ein erneuter Aufruf desselben „Ankers“ würde keine Änderung ergeben (kein *reload*), was eine mehrmalige Sortierung (absteigend/aufsteigend im direkten Wechsel) unmöglich macht. Die hier vorliegende Hilfsfunktion realisiert nun nach Übergabe des betroffenen Parameters mittels JavaScript die Funktionalität eines Ankersprungs.

...

```
window.location.href = ae;
```

...

Damit wird neben der gewollten Fokussierung auf die entsprechende Tabelle auch der hier erforderliche Neuaufruf der Seite bei jeder (zwecks Sortierung) erfolgten Betätigung eines Hyperlinks gewährleistet. Beachtet werden muss allerdings, dass hier eine direkte Parameterübergabe inklusive Rautenzeichen Aufgrund der Verwechslung mit einem „normalen“ HTML Anker nicht möglich ist - das Zeichen muss einfach vor der Verarbeitung in dieser Funktion ergänzt werden.

(2) MfzMessageLib.asp

Für das automatische Versenden von Mitteilungen im Rahmen der Dienstnutzung und für das Verschicken von Mitteilungen an einen anderen Nutzer über das entsprechende Formular werden einige speziell für ihren jeweiligen Zweck erstellte Funktionen benötigt, welche in dieser diensteigenen Bibliothek zusammengefasst sind. Aus praktischen Gründen werden diese Funktionen eingesetzt, um nicht in jedem Fall der Anwendung den entsprechenden Quelltext von neuem einfügen zu müssen.

- `mail_message($id)` → Zum Versenden einer einfachen Nachricht über die Seite *formular.MfzMessage.asp* wird diese Funktion verwendet. Neben dem standardmäßigen Abruf von Name, Vorname und E-Mailadresse des Empfängers wird die eingegebene Nachricht nur noch mit dem vorgegebenen allgemeinen Textteil und der passenden Betreffszeile zusammen in die PHP – Mailfunktion eingefügt und automatisch verschickt.
- `mail_gebucht($id, $iid, $freq_aktuell_lang)` → Wurde bei einem selbst eingetragenen Angebot oder Gesuch eine Buchung vorgenommen, wird der Nutzer durch eine automatische Mitteilung (realisiert über diese Funktion) darauf aufmerksam gemacht. In diesem Fall ist das komplette Textgerüst der Mail bereits vorgegeben. Individuell werden vor dem Abschi-

cken der jeweiligen Mail noch die benötigten Informationen aus der Datenbank abgerufen, um die entsprechenden Variablen Im Text mit den passenden Werten zu füllen. Einzig die Frequenzanzeige wird bereits im Buchungsformular gebildet und übergeben (kein einfacher Abruf möglich).

- `mail_buch_update($id, $iid, $start_old, $ziel_old, $dat_old, $freq_old, $freq_old_lang, $plaetze_old)`
→ Falls eine bestehende Buchung von dem betreffenden Nutzer in ihren Details geändert (nicht gelöscht) wird, kommt diese Funktion zum Einsatz. Zur besseren Übersicht werden bei den jeweils geänderten Parametern der alte und der neue Wert gegenübergestellt. Neben dem einfachen Datenbankaufruf zur Abfrage der aktuellen Parameter muss die aktuelle Frequenz für die Darstellung noch entsprechend aufgearbeitet werden (Anzeige falls nötig)
- `mail_buch_cancel($id, $start, $ziel, $datum, $frequenz_lang_mail, $art_inserat)` → Wird eine Buchung zurückgezogen, erfolgt simple Mitteilung anhand dieser Funktion. Zur Identifikation werden die Schlüsselparameter Art, Start-/Zielort, Datum und Frequenz direkt übergeben und in den Textteil eingebaut, und müssen nicht extra innerhalb dieser Funktion abgerufen werden. Die entsprechende Zusammensetzung des Frequenzstrings erfolgt demnach bereits auf der Seite zum Löschen der Buchung (*MfzBuchungsDelete.asp*).
- `mail_inserat_update($id, $iid, $dat_old, $dat, $frequenz, $uhrzeit, $uhrzeit1, $start, $sue1, $sue2, $sue3, $ziel, $treff, $platz, $preis, $preis1, $bem)`
→ Nimmt ein Nutzer nichtkritische bzw. kompatible Änderungen an einem seiner Angebote bzw. Gesuche vor, wird jedem Nutzer der dieses Inserat gebucht hat, eine entsprechende Mitteilung geschickt. Ähnlich zum Update einer Buchung, werden hier die abgeänderten Details in ihrem alten und neuen Zustand gegenübergestellt. In diesem Fall ist die Fülle an Parametern jedoch größer. Diverse Einzelheiten verlangen dabei eine Nachbearbeitung (im Gegensatz zu einer simplen Gegenüberstellung.) bevor sie korrekt bzw. sinnvoll angezeigt werden können (siehe Hinzufügen der Bemerkung „keine Angabe“ bei Nichteintragung eines Zwischenhaltepunktes).
- `mail_inserat_update_achtung($id, $iid, $art_achtung, $datum_achtung, $frequenz_achtung, $startort_achtung, $ueber1_achtung, $ueber2_achtung, $ueber3_achtung, $zielort_achtung, $platz_achtung, $count_buch)` → Möchte ein Nutzer jedoch Änderungen vornehmen, die aufgrund bestehender

Buchungen nicht möglich sind, wird den Personen, die das Inserat gebucht haben eine Hinweismail geschickt. Darin sind die betroffenen Details zu finden, die eigentlich geändert werden sollen. Hierbei erfolgt wieder eine Gegenüberstellung der relevanten Parameter. Diese Mail ist als Hinweis für die buchenden Nutzer zu deuten, sich so bald wie möglich mit dem Inserenten wegen möglicher Konsequenzen in Verbindung zu setzen.

- `mail_inserat_cancel($id, $art, $start, $ziel, $datum)` → Wird ein Angebot oder Gesuch gelöscht, wird im Falle bestehender Buchungen an die betreffenden Personen eine dementsprechende Mitteilung geschickt. Analog zum Löschen einer Buchung werden hier zur Identifikation der Fahrt der allgemeine Start- und Zielort, sowie das Fahrt- bzw. Startdatum (bei regelmäßigen Fahrten) angeführt. Auf die Frequenz wird wegen möglicher Unterschiede bei der Buchung verzichtet.
- `mail_bewertung_ab($id, $typ, $art, $start, $ziel, $datum, $freq_other, $updateopt)` → Hat ein Nutzer seine Bewertung zu einer bestimmten Fahrt abgegeben, wird der zugehörige Inserent per E-Mail darauf aufmerksam gemacht. Es werden zwei Situationen unterschieden: erstmalige oder nachfolgende Bewertung (bei regelmäßigen Fahrten). Im zweiten Fall ist der Mitteilungstext jedoch nur leicht abgeändert. Ansonsten erfolgt die Standardverarbeitung einer solchen Funktion zum Versenden einer Nachricht.

- **individuell modifizierte Klassen**

- (1) `FormObjectMfz.asp`

Diese Datei stellt eine speziell für den Mitfahrdienst modifizierte Variante von *FormObject.asp* dar. Die Standardversion von `RenderControl($obj)` passt nicht in jeder Situation zur jeweils geforderten Umsetzung. Deshalb wurden für den Mitfahrdienst mehrere Funktionen nach dem Vorbild von `RenderControl($obj)` erstellt, um bestimmte individuelle Aufgaben erfüllen zu können oder stellenweise unpassende Details auszubessern.

- `DisableElementCheck($obj)` → Mit Zuweisung dieser Funktion auf eine Checkbox wird veranlasst, dass bei Betätigung dieser Checkbox ein bestimmtes Element (im Wechsel) aktiviert bzw. deaktiviert wird. Jedes betroffene (d.h. zu ändernde) Element muss dabei innerhalb der Funktion separat aufgeführt werden, da die jeweils anzuwendende Funktion aus *lib_mfz.asp* variieren kann (aufgrund verschiedener Objekt-/Elementtypen und Szenarien der Anwendung). Bei Formularen die in einem Speichervorgang resultieren können, oder bei denen zumindest im Formularablauf vor der endgültigen Bearbeitung ein *post*- Vorgang möglich/vorgesehen ist (wie etwa bei der Eintragung von Inseraten), muss für die betroffenen

Checkboxen zusätzlich die Funktion `changecheckbox()` eingebunden werden. Damit wird die korrekte Weiterverarbeitung dieser Elemente sichergestellt.

- `RenderSelectDisabled($obj)` → In manchen Fällen muss eine Auswahlliste beim Aufruf einer Formularseite als „disabled“ (d.h. deaktiviert) geladen werden (siehe optionaler Toleranzbereich bei der Detailsuche). Bei den betroffenen Elementen muss daraufhin diese Funktion zur entsprechenden Umsetzung des Objektes zugewiesen werden. Wegen der Umstellung auf *FormObjectMfz* ist die Definition der jeweiligen Listenelemente (`$this->options[...]`) nicht mehr in der eigentlichen Definitionsdatei des Formulars, sondern hier in *FormObjectMfz.asp* zu finden).
- `RenderSelect($obj)` → Mit dieser Funktion wird eine Auswahlliste in ihrem Normalzustand, das heißt aktiv, umgesetzt. Durch `RenderControl($obj)` ist dieser Aspekt zwar schon abgedeckt, jedoch löste in der zu diesem Zeitpunkt vorliegenden Version jede Auswahl (d.h. jeder Klick) ein Absenden des Formulars aus. Das unterbricht nur ungewollt (und sinnlos) die Abarbeitung dieses Formulars, weswegen eine angepasste Variante (zumindest in diesem Fall) implementiert werden musste.
- `RenderInputDisabled($obj)` → Für den Fall, dass ein Textzeilenelement beim Laden der Seite oder durch eine entsprechende Aktion veranlasst inaktiv geschaltet werden soll, wird diese Funktion zur Umsetzung des Elements eingebunden (bspw. wenn ein Gesuch vorliegt bzw. als Folge der Aktivierung der Funktion `disable_stop()` nach Auswahl der Art Gesuch).
- `RenderInputEnabled($obj)` → Bei Textzeilen, die aufgrund einer möglichen Interaktion (wieder) inaktiv geschaltet werden könnten (wie etwa die Zwischenhaltepunkte), wird zur besseren Unterscheidung und zur Vermeidung von Komplikationen bei der Realisierung der standardmäßigen, eingabebereiten Zeile diese Funktion hier verwendet.
- `RenderInputChangeRadio($obj)` → Im detaillierten Suchformular können Start- bzw. Zielort per Hand eingegeben werden oder man wählt einen voreingestellten Wert (Mittweida/Rosswein). Nutzt man die Auswahl per Radiobutton und entscheidet sich aber danach doch für eine manuelle Eingabe, wird bei dem Klick auf die Textzeile die Markierung des Radiobutton wieder entfernt. Mit Hilfe dieser Funktion erhält das standardmäßige Input-Element diese Fähigkeit.
- `RenderInputInhaltGrey($obj, $input_inhalt)` → Bei der einfachen Suche auf der Startseite des Mitfahrdienstes sind die Eingabefelder mit einem grauen Definitionstext vorbelegt. Mit-

tels dieser Funktion werden die einfachen Textzeilen realisiert (alles außer Datum). Klickt ein Nutzer in die Zeile, verschwindet der voreingestellte Inhalt, und die Textfarbe für die folgende Eingabe wird auf schwarz geändert.

- `RenderDateInhaltGrey($obj, $input_inhalt)` → Im Gegensatz zur vorherigen Funktion wird hierbei eine Textzeile zur Datumseingabe realisiert.
- `RenderCheckEnabled($obj)` → Soll eine einfache Checkbox realisiert werden, die keine Interaktion mit anderen Objekten ermöglicht, wird diese Funktion benötigt.
`RenderControl($obj)` reicht hierbei nicht aus, da die Funktion `changecheckbox(cbl)` eingearbeitet werden musste.
- `RenderCheckGruppe($obj)` → Bei den Formularen für die Buchung eines Angebots und der Eintragung des persönlichen Profils sind jeweils mehrere als Gruppe angeordnete Checkboxes zu finden (zur Auswahl von einzelnen Tagen bei täglichen/werktäglichem Angeboten bzw. zur Auswahl nutzbarer Sprachen). Im Gegensatz zur Eintragung von Inseraten oder der Detailsuche wird hier eine reine Auswahl - und nicht zusätzlich das An- und Ausschalten diverser Formularelemente - bezweckt, weswegen die Realisierung dieser Checkboxes zur besseren Trennung beider Eigenschaften über eine separate Funktion gehandhabt wird. Hauptsächlich soll hierbei die korrekte Verarbeitung und (im Falle des Buchungsformulars auch) die Aufrechterhaltung der Parameterwerte nach einem post - Vorgang des Formulars sichergestellt werden.
- `RenderKommentar80($obj)` → Für den Bewertungskommentar soll eine Beschränkung der Länge gelten, das heißt (ähnlich zur Bewertungsfunktion bei *eBay*) sollen für diese kurze Bemerkung nur maximal 80 Zeichen zugelassen werden. Diese Beschränkung existiert in der Standardvariante für ein *input*-Element jedoch nicht, weshalb dieser Mangel mittels der hier angegebenen Funktion umgangen wird.
- `SelectInhaltPlatz($obj, $plaetze_frei)` → Bei der über die Seite *formular.MfzBuchung.asp* getätigten Reservierung von Sitzplätzen ist diese Funktion verantwortlich, aus der (für die ausgewählte Frequenz) übergebenen Anzahl freier Plätze eine Auswahlliste zu generieren. Stehen beispielsweise drei Sitzplätze zur Verfügung, wird eine Liste mit den Auswahloptionen 1,2 und 3 erstellt, wobei der Nutzer dann die Möglichkeit hat einen oder gleich mehrere Plätze auf einmal zu buchen.
- `SelectInhaltStart($obj, $startort, $ueber1, $ueber2, $ueber3)` → Ähnlich zur vorherigen Funktion wird hierbei (passend zu den übergebenen Parametern) eine Aus-

wahlliste erstellt. Allerdings handelt es sich hierbei um die Liste der möglichen Startorte einer Buchung. Im Gegensatz zu den Zielorten, ist diese Liste während des betroffenen Buchungsvorgangs jedoch immer gleich bleibend. Sollten keine oder weniger als die drei möglichen Zwischenhalte definiert sein, wird die betroffene Option einfach nicht für die Liste berücksichtigt.

- `SelectInhaltZiel($obj, $zielort, $ueber11, $ueber21, $ueber31)` → Diese Funktion ist analog zur Vorherigen beim Buchungsvorgang für die Auflistung von Orten, hier von möglichen Zielorten, verantwortlich. Diese List passt sich dabei allerdings auf den gewählten Startort an, um eine Übereinstimmung von Start- und Zielort zu vermeiden (nur relevant bei eingetragenen Zwischenhaltepunkten).

(2) FormObjectUhrzeitMfz.asp

Analog zu *FormObjectTel.asp* und *FormObjectDate.asp* beinhaltet diese Datei ein vordefiniertes Schema eines speziellen Objektes, in diesem Fall zur Angabe einer Uhrzeit. Jedoch wurde hier die ursprüngliche Datei modifiziert, damit nur noch für eine Uhrzeit gültige Zahlenwerte zugelassen sind.

(3) FormObjectDateMfz.asp

Diese Datei ist im Grunde identisch zu *FormObjectDate.asp*. Jedoch wird eine Anpassung für *FormObjectMfz* als Basisklasse benötigt, sobald eine Textzeile Zur Datumseingabe nicht über die Standardfunktion `RenderControl($obj)` generiert wird (hier bei der einfachen Suche auf der Startseite der Fall).

▪ globale Stylesheets (Standardvarianten)

(1) styles.css

Für die genutzte Programmierungsumgebung (*tc-teach*) stellt diese Datei einen zentralen global verfügbaren Stylesheet dar, welcher universell einsetzbar ist. Es existieren zahlreiche Festlegungen sowohl für allgemeine Elemente, als auch für einige spezielle Anwendungsbereiche.

(2) forms.css

Dieser Stylesheet enthält wichtige Elementbeschreibungen, die bei der Programmierung von Formularen für deren korrekte Darstellung benötigt werden (im Sinne der Einheitlichkeit des Erscheinungsbildes).

▪ (teils) modifizierte lokale Stylesheets

(1) MfzStyles.css

Dieser individuell für den Mitfahrdienst erstellte Stylesheet enthält (teils in leicht abgeänderter Form) einen Teil der Elemente der lokalen Datei *styles.css* aus dem Projekt der Wohnungsbörse, sowie mehrere Elemente, die bereits in den globalen Standarddateien zu finden sind (und für die Nutzung an dieser Stelle modifiziert wurden). Die Übernahme aus der Wohnungsbörse ist damit zu erklären, dass sich der Mitfahrdienst optisch an diesem Projekt orientieren sollte. Außerdem hat sich bei der Entwicklung der Seiten gezeigt,

dass die Anpassung einiger Definitionen nötig war, um das Erscheinungsbild in der gewünschten Form realisieren zu können.

3.3 Datenbankstruktur

Im folgenden Abschnitt wird die für diesen Dienst zugrunde liegende Datenbankstruktur erläutert. Das heißt, die verwendeten Tabellen werden inhaltlich bzw. strukturell beschrieben und deren Zusammenhänge aufgezeigt, wobei jeweils die wichtigsten Parameter hervorgehoben werden.

Zur Erstellung der Tabellen und deren Administration wurde in diesem Fall die Applikation „*phpMyAdmin*“ genutzt. Über deren einfach zu bedienendes grafisches Web-Interface ist es sowohl im WYSIWYG Verfahren, als auch mittels manueller Eingabe von SQL Befehlen möglich, diverse Operationen auszuführen, wie etwa das Anlegen und Löschen von Tabellen oder deren Modifikation. Außerdem können detaillierte Einstellungen bzw. Parametereigenschaften festgelegt werden, beispielsweise den jeweils zu verwendenen Datentyp.

■ Gesamtstruktur

Insgesamt werden sieben individuell für diesen Mitfahrdienst erstellte Tabellen benötigt. Die nachfolgende Grafik soll dabei die bestehenden Zusammenhänge veranschaulichen:

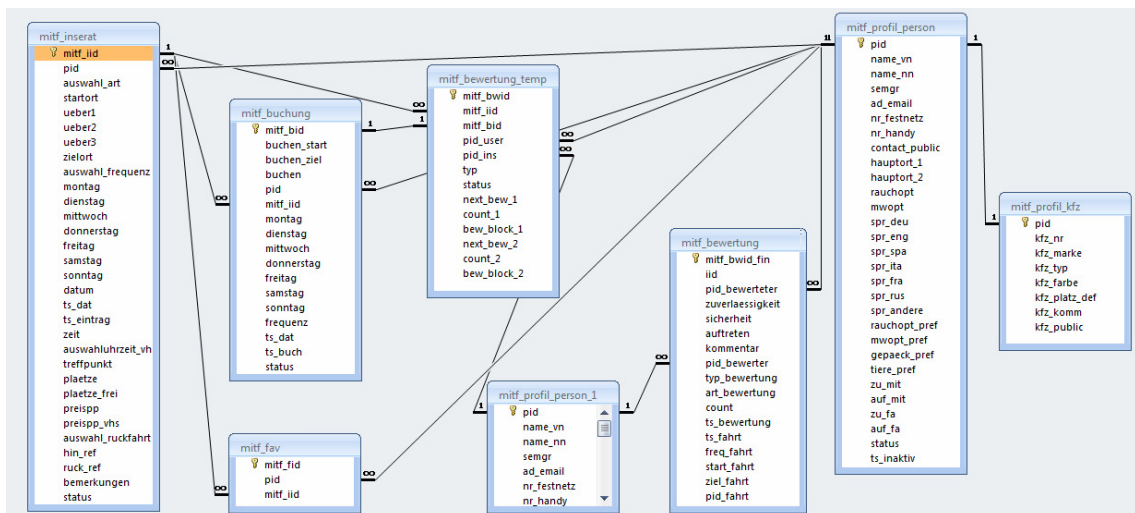


Abbildung 3-2 Screenshot Tabellenverknüpfungen (Microsoft Access 2007)

Die Verbindungen der Tabellen untereinander werden als Relationships bezeichnet. Dabei sind hier grundsätzlich zwei verschiedene Typen vertreten – die 1:n und die 1:1 Beziehung. Der Typ 1:n existiert beispielsweise zwischen der Inserats- und der Buchungstabelle. Das bedeutet: jedes Angebot oder Gesuch ist als eigenständig zu betrachten und muss eindeutig identifizierbar sein. Deshalb darf es beim zugehörigen Index *mitf_iid* (dem Primärschlüssel der Tabelle) keine Duplikate geben. Dieser Index stellt auch die Verbindung mit den zugehörigen Buchungseinträgen dar (ist dort also der Fremdschlüssel). Da Angebote durchaus von mehreren Personen gebucht werden können, darf in der betroffenen Tabelle die Inseratskennung jedoch mehrfach erscheinen. Dieselbe Beziehung besteht außerdem noch zur Tabelle für temporäre Bewertungsdatensätze, sowie der Favoritenliste. Weiterhin ist die eindeutige Identifikation jedes Nutzers eine der Grundvoraussetzungen für die Funktionalität und Anwenderfreundlichkeit des Mitfahrdienstes. Wegen des einmaligen Intranetlogins kann sich eine Person im Gegensatz zu anderen Diensten dieser Art hier nicht einfach unter einem anderen Namen neu anmelden.

den. Deshalb gibt es für jeden Nutzer auch nur einen Profileintrag, welcher jedoch vorhanden sein muss, um die weitergehenden Funktionen des Dienstes nutzen zu können. Diese eindeutige Nutzerkennung steht auch in einer 1:n Beziehung mit allen anderen Tabellen (außer dem Kfz-Profil), da jeder Nutzer natürlich eine Vielzahl an Inseraten eintragen, Buchungen vornehmen oder Bewertungen erhalten kann. Jedoch darf jede Person neben dem persönlichen auch nur ein Fahrzeugprofil besitzen, weswegen zwischen diesen beiden Tabellen eine 1:1 Beziehung existiert. Eine weitere Relation dieser Art ist zwischen den Buchungen und den dazu gehörigen temporären Bewertungseinträgen zu finden, da zwar mit einem Buchungsvorgang zwei potentielle Bewertungen verbunden sind (Fahrer und Mitfahrer), aber beide in einem Datensatz verpackt werden.

- **Tabelle des persönlichen Profils – mitf_profil_person**

Datenbank portal - Tabelle mitf_profil_person

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/>	pid	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	name_vn	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	name_nn	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	semgr	varchar(10)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ad_email	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	nr_festnetz	varchar(25)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	nr_handy	varchar(25)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	contact_public	varchar(4)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	hauptort_1	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	hauptort_2	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	rachopt	varchar(4)		Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	mwopt	char(1)		Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_deu	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_eng	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_spa	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_ita	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_fra	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_rus	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	spr_andere	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	rachopt_pref	varchar(15)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	mwopt_pref	varchar(15)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	gepaeck_pref	varchar(15)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	tiere_pref	varchar(15)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	zu_mit	float(4,3)	UNSIGNED	Nein	0.000		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auf_mit	float(4,3)	UNSIGNED	Nein	0.000		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	zu_fa	float(4,3)	UNSIGNED	Nein	0.000		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auf_fa	float(4,3)	UNSIGNED	Nein	0.000		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	status	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ts_inaktiv	int(10)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY	PRIMARY	7	Löschen Bearbeiten	pid
pid	UNIQUE	7	Löschen Bearbeiten	pid

Abbildung 3-3 Screenshot Tabelle des persönlichen Profils

Will ein Nutzer aktiv am Mitfahrdienst teilnehmen, muss dessen persönliches Profil eingetragen sein. Das heißt, wenigstens die als Pflichtangaben definierten Parameter wie Name und E-Mail Adresse sind anzugeben. Der bestehende Intranet-login wird als Primärschlüssel *pid* übernommen, wobei Duplikate nicht zugelassen sind und jedem Nutzer somit ein Datensatz eindeutig zugeordnet werden kann. Außer den größtenteils freiwilligen Profilangaben sind in dieser Tabelle aus praktischen Gründen außerdem noch die nach Art und Parameter getrennten Durchschnittswerte aller bisherig für diesen Nutzer abgegebenen Bewertungen gespeichert.

- **Tabelle des Fahrzeugprofils – mitf_profil_kfz**

Datenbank *portal* - Tabelle *mitf_profil_kfz*

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/>	<u>pid</u>	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_nr	varchar(25)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_marke	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_typ	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_farbe	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_platz_def	tinyint(1)	UNSIGNED	Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_komm	varchar(200)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kfz_public	varchar(4)		Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY	PRIMARY	3	Löschen Bearbeiten	pid
pid	UNIQUE	3	Löschen Bearbeiten	pid

Abbildung 3-4 Screenshot Tabelle des Fahrzeugprofils

Der Parameter *pid* bildet auch in diesem Fall den Primärschlüssel der Tabelle und stellt den Intranetlogin der betreffenden Person dar. Genauso ist auch hier nur ein Datensatz pro Nutzer möglich. Jedoch ist im Gegensatz zu den persönlichen Angaben dieses Profil als komplett freiwillig zu betrachten.

- **Tabelle der Inserate – mitf_inserat**

Datenbank *portal* - Tabelle *mitf_inserat*

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/>	mitf_iid	mediumint(8)	UNSIGNED	Nein		auto_increment	Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auswahl_art	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	startort	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ueber1	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ueber2	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ueber3	varchar(50)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	zielort	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auswahl_frequenz	varchar(15)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	montag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	dienstag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	mittwoch	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	donnerstag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	freitag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	samstag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	sonntag	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	datum	varchar(10)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ts_dat	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ts_eintrag	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	zeit	varchar(5)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auswahluhrzeit_vhs	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	treffpunkt	varchar(80)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	plaetze	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	plaetze_frei	tinyint(1)	UNSIGNED	Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	preispp	smallint(3)		Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	preispp_vhs	tinyint(1)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auswahl_ruckfahrt	varchar(4)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	hin_ref	mediumint(8)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ruck_ref	mediumint(8)	UNSIGNED	Ja	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	bemerkungen	varchar(200)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	status	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY	PRIMARY	15	Löschen Bearbeiten	mitf_iid
mitf_iid	UNIQUE	15	Löschen Bearbeiten	mitf_iid

Abbildung 3-5 Screenshot Tabelle der Inserate

Diese Tabelle setzt sich zum einen aus den Parametern zusammen, welche mit den entsprechenden Objekten aus dem Formular zur Eintragung von Inseraten korrespondieren. Für die weitere Verarbeitung innerhalb des Dienstablaufs werden aber noch ein paar zusätzliche Parameter benötigt - Timestamp Angaben des Inserats- und des Eingabedatums, die benötigten Parameter zur Anzeige einer Rückfahrt, sowie der aktuelle Status, um zu zeigen ob der betreffende Eintrag aktiv (= buchbar) oder inaktiv ist (= abgelaufen/gelöscht, Daten aber noch für Bewertung benötigt). Außerdem wird je nach gewählter Frequenz die Platzanzahl auf die einzelnen Wochentage aufgeteilt, um für die Buchungsfunktion eine Platzverwaltung realisieren zu können. Ist ein Inserateintrag abgelaufen, oder wurde vom Inserenten gelöscht, wird dieser entweder sofort aus der Datenbank entfernt, oder bei ausstehenden Bewertungen solange aufrechterhalten, bis diese vorgenommen wurden. Der Primärschlüssel ist in diesem Fall die Inserats- ID *mitf_iid*. Der Parameter *pid* ist dabei als Fremdschlüssel vorgesehen, um die personenbezogene Verbindung zu anderen Tabellen/Datensätzen herstellen zu können.

- **Tabelle der vorgemerkten Inserate (Favoritenliste) – mitf_fav**

Datenbank *sportal* - Tabelle *mitf_fav*

Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/> mitf_fid	mediumint(8)	UNSIGNED	Nein		auto_increment	Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> pid	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> mitf_iid	mediumint(8)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY	PRIMARY	0	Löschen Bearbeiten	mitf_fid
mitf_fid	UNIQUE	0	Löschen Bearbeiten	mitf_fid

Abbildung 3-6 Screenshot Tabelle der vorgemerkten Inserate

Diese Tabelle ist nicht für konkreten eigenständigen Inhalt vorgesehen, denn die Einträge dienen nur als Verweise auf die betroffenen Inserate. Der wichtigste Parameter ist deshalb der Fremdschlüssel *mitf_iid*. Aber auch hier muss jeder Eintrag einem bestimmten Nutzer zugeordnet werden, weswegen der Parameter *pid* auch hier enthalten ist.

- **Tabelle der vorgenommenen Buchungen – mitf_buchung**

Datenbank *sportal* - Tabelle *mitf_buchung*

Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/> mitf_bid	mediumint(8)	UNSIGNED	Nein		auto_increment	Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> buchen_start	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> buchen_ziel	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> buchen	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> pid	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> mitf_iid	mediumint(8)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> montag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> diensttag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> mittwoch	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> donnerstag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> freitag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> samstag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> sonntag	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> frequenz	varchar(15)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> ts_dat	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> ts_buch	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/> status	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY	PRIMARY	26	Löschen Bearbeiten	mitf_bid
mitf_bid	UNIQUE	26	Löschen Bearbeiten	mitf_bid

Abbildung 3-7 Screenshot Tabelle der vorgenommenen Buchungen

Neben den Details, welche vom Nutzer direkt während des Buchungsvorganges beeinflusst werden (Start-/Zielort, Platzanzahl, Frequenz), sind auch hier wie bei der Inseratstabelle zur nachfolgenden Verarbeitung bzw. Anzeige zusätzliche Parameter enthalten. Die gebuchte Anzahl Sitzplätze wird je nach Frequenz auf die entsprechenden Wochentage aufgeteilt, weiterhin werden Buchungsdatum und – Status ergänzt. Der Parameter *ts_dat* stellt in diesem Fall den ersten Geltungstag der Buchung, das heißt den ersten (möglichen) Tag der Fahrt, dar (wird später etwa im Zusammenhang mit der Bewertungsfreigabe benötigt). der Primärschlüssel ist *mitf_bid*, und über *mitf_iid* wird die Verbindung mit den zugehörigen Inserateinträgen hergestellt.

- **Tabelle der temporären Bewertungsdatensätze – mitf_bewertung_temp**

Datenbank portal - Tabelle mitf_bewertung_temp

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/>	mitf_bwid	mediumint(8)	UNSIGNED	Nein		auto_increment	Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	mitf_iid	mediumint(8)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	mitf_bid	mediumint(8)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid_user	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid_ins	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	typ	varchar(10)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	status	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	next_bew_1	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	count_1	smallint(5)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	bew_block_1	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	next_bew_2	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	count_2	smallint(5)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	bew_block_2	tinyint(1)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [Dokumentation](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY PRIMARY		30	Löschen Bearbeiten	mitf_bwid
mitf_bwid UNIQUE		30	Löschen Bearbeiten	mitf_bwid

Abbildung 3-8 Screenshot Tabelle der temporären Bewertungsdatensätze

Nach einer vorgenommenen Buchung wird der zugehörige temporäre Datensatz erstellt, indem sowohl der Bewertungsvorgang für den Buchenden selbst (*pid_user/1*), als auch dem Inserenten (*pid_ins/2*) angelegt ist. Grundlegend sind hierbei die beiden *next_bew* Parameter, welche Aussagen, ab wenn die (nächste) Bewertung freigegeben ist (regelmäßige Fahrten können mehrfach bewertet werden). Der Wert *bew_block* ist wichtig im Falle der Buchung von Hin- und Rückfahrt. Normalerweise wird der temporäre Datensatz gelöscht, sobald ein Bewertungsvorgang abgeschlossen ist (etwa um Doppelbewertungen zu verhindern). Bei regelmäßigen (noch aktiven) Fahrten bleiben die Daten aber so lange bestehen, bis die zugehörige Buchung gelöscht wurde. Außerdem muss ein echtes Update der Bewertung möglich sein, um nicht für jeden einzelnen Vorgang einen neuen Datensatz zu erstellen. Weiterhin sind in diesem Fall die Buchungsnummer (*mitf_bid*) und die zugehörige Inseratsnummer (*mitf_iid*) als Fremdschlüssel vorgesehen.

- **Tabelle der endgültigen Bewertungsdatensätze – mitf_bewertung**

Datenbank *portal* - Tabelle *mitf_bewertung*

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion					
<input type="checkbox"/>	mitf_bwid_fin	mediumint(8)	UNSIGNED	Nein		auto_increment	Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	iid	mediumint(8)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid_bewerteter	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	zuverlaessigkeit	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	sicherheit	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	auftreten	tinyint(1)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	kommentar	varchar(100)		Ja	NULL		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid_bewerter	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	typ_bewertung	varchar(10)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	art_bewertung	varchar(8)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	count	smallint(5)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ts_bewertung	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ts_fahrt	int(10)	UNSIGNED	Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	freq_fahrt	varchar(50)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	start_fahrt	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	ziel_fahrt	varchar(50)		Nein			Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext
<input type="checkbox"/>	pid_fahrt	varchar(8)		Nein	0		Ändern	Löschen	Primärschlüssel	Index	Unique	Volltext

Indizes : [\[Dokumentation\]](#)

Name	Typ	Kardinalität	Aktion	Feld
PRIMARY PRIMARY		58	Löschen Bearbeiten	mitf_bwid_fin

Abbildung 3-9 Screenshot Tabelle der endgültigen Bewertungsdatensätze

Neben den konkreten Parametern der Bewertung selbst werden für die Anzeige der Details noch ein paar wenige Einzelheiten des zugehörigen Inserats mit abgespeichert. Da nach abgeschlossener Bewertung keine direkte Verbindung mehr zum Inseratseintrag besteht und dieser nach Ablauf seiner Gültigkeit (oder wenn er gelöscht wurde) sowieso nicht mehr verfügbar ist, sind diese Einzelheiten sinnvoll, um außer den Bewertungsdetails auch noch etwas zum Kontext der Fahrt sagen zu können. Der Parameter *iid* ist nur für mögliche Bewertungsupdates relevant, ansonsten ist dieser mit dem Wert 0 belegt, um später keine unerlaubten Bewertungsvorgänge zu erlauben.

4 Programmtechnische Umsetzung

4.1 Anmerkungen zur objektbasierten Programmierung

Durch die Erweiterung des Studentenportals und die zunehmende grundlegende Angleichung der Service bzw. Informationsseiten wurde bei der Programmierung dieser Version eines Mitfahrdienstes speziell Wert darauf gelegt, das übergeordnete Design (und damit die maßgebliche Programmiergrundlage selbst) am derzeit gängigen Standard auszurichten. Damit ergab sich konsequenterweise die grundlegende Verwendung der objektbasierten Realisierung - im wesentlichsten bei den Formular- und Detailseiten - im Gegensatz zu komplett selbst zusammengesetzten Elementen und dem eher eigenständigen und damit weniger optimierten Einsatz von Funktionen und Datenbankabfragen.

Um eine logische Trennung und eine bessere Übersichtlichkeit zu erreichen, werden Objektdefinition und Implementierung auf separate Seiten verteilt. Beispielsweise bei der Eintragung bzw. Änderung von Inseraten, sowie deren Darstellung stellt *MfzInserat.asp* die Definitionen bereit, wobei sich die konkrete Realisierung des Formulars in der Datei *formular.MfzInserat.asp* befindet, und die Darstellung der Details auf der Seite *MfzInseratDetails.asp* zu finden ist.

Die einzelnen Elemente werden über Parameter wie Typ (*input*, *select*, *checkbox* etc.), Name oder Optionen inhaltlich definiert, und Angaben zu Beschreibung (*info*) oder style beeinflussen deren Darstellung. Aber auch ein bestimmtes Gültigkeitsmuster (*valid*) oder ein fester Modus (Pflichtangabe, read-only) kann direkt in der Objektbeschreibung festgelegt werden. Bei den Detailseiten ist der Typ eines Elements im Grunde irrelevant, da in diesem Fall nur eine Textdarstellung von Bedeutung ist.

Entweder werden diese Objekte bei Formularen für einen neuen Eintrag leer initialisiert, oder bei Übergabe einer ID mit den entsprechenden Inhalten aus der Datenbank gefüllt. Die einzelnen Elemente können auch durch separate Datenbankabfragen vorgelegt werden.

Alle Definitionen befinden sich innerhalb einer Klasse, welche auf der Formularseite entsprechend zugewiesen werden muss (vom Ablauf her nach dem *include* der Definitionsdatei). Des Weiteren werden benötigte Programmbibliotheken einbezogen oder auch diverse Datenbankabfragen durchgeführt, um verschiedene Parameter zur späteren Weiterverarbeitung abzurufen oder angepasste Mitteilungen anzuzeigen. Alle Objekte die letztendlich verwendet werden sollen, müssen an ihrem Bestimmungsort innerhalb des HTML-Teils des Quelltextes noch konkret implementiert werden. Im Fall der Formulare entscheidet dabei die jeweils angewendete Funktion (bspw. `RenderControl($obj)`) über die spezifische Umsetzung des Objektes in HTML-Text. Innerhalb dieser Funktion kann der letztendlich zu parsende Quelltext durch die Verwendung von Variablen und deren möglicher Manipulation noch beeinflusst werden. Dafür müssen diese Funktionen aber in manchen Fällen individuell angepasst (oder neu erstellt) werden (siehe `RenderInputDisabled($obj)`), da deren Standard Varianten eher die allgemeine Realisierung abdecken.

Die Objekte können auch dazu verwendet werden Informationen oder Mitteilungen anzuzeigen. Dabei werden zwei Situationen unterschieden: Die durch ein *mouse-over* über das Label zu erreichenden Hilfetexte...

...

```
$obj->props[Strecke][startort]->RenderLabel()
```

oder

```
$LabelStartort->RenderLabel()
```

...

... und die Fehlermeldungen nach der (negativen) Eingabeprüfung:

...

```
$obj->props[Strecke][startort]->RenderInfo()
```

...

Grundlegend ist der Inhalt beider Meldungen identisch, und wird als Parameter Info der jeweiligen Objektdefinition festgehalten. Lediglich die unterschiedliche Verarbeitung (Label oder Info) entscheidet über die letztendliche Darstellung (grün unterlegt = einfacher Hilfetext; rot unterlegt = Warnhinweis). Werden die Objekte jedoch auf unterschiedlichen Seiten verwendet (wie etwa bei Formularen und Detailseiten), dann kann es sein, dass der Informationstext nicht in allen Situationen passend ist. In so einem Fall können auch separate Label in der Form `$LabelStartort=new FormObject(...)` definiert und dann an der entsprechenden Stelle eingefügt werden. Individuelle Meldungen (außerhalb der Objektdefinition), vor allem Fehlermeldungen, sind selbstverständlich auch möglich.

Der Teil des PHP Quellcodes, welcher sich nach der Variable `$post` befindet, wird ausschließlich nach dem Abschicken des Formulars abgearbeitet (= Eingabeprüfung und Speichern). Dabei können aufgrund verschiedener Ablaufszenarien auch mehrere Bearbeitungswege existieren.

Die Abspeicherung der Formulardaten erfolgt anhand der Verwendung eines Datenbankobjektes (`DbObject`), erstellt in der Definitionsdatei. Alle zwischen dem Formular und der angegebenen Tabelle der betreffenden Datenbank korrespondierenden Elemente werden automatisch in den Datensatz übernommen, wobei der Name des jeweiligen Objektes ausschlaggebend ist. Sollen bestimmte Variablen vor der Speicherung noch verarbeitet, oder individuell ergänzte (nicht objektbasierte) Elemente mit einbezogen werden, ist dies durch eine simple Zuweisung problemlos möglich (bspw. `$this->auswahl_art=$_POST[R1]`).

4.2 Anmerkungen zur Programmierung der Übersichtsseiten

Die verschiedenen im Mitfahrdienst verwendeten Übersichtsseiten sind grundsätzlich zweigeteilt programmiert. Das heißt, die einzelnen konkreten mit Daten gefüllten Tabellen sind jeweils auf einer separaten Seite ausgelagert. Damit bleibt das „Gerüst“ der Seite in einem übersichtlichen Zustand, und die benötigten Tabellen müssen nur noch implementiert werden. Hier als Beispiel die Implementierung der Angebotstabelle in der Übersicht für Angebote und Gesuche (*MfzInseratUebersicht.asp*):

...

```
$page->CheckBarCss("$anzahl_angebote Angebot(e) eingetragen  
($vn_pid_user. $nn_pid_user)", "zeigean", $zeigean);  
if ($zeigean=="checked") include("MfzInseratAngebot.asp");
```

...

(über `$vn_pid_user` und `$nn_pid_user` wird der Name des Inserenten dargestellt)

In diesem Fall beinhaltet die Seite *MfzInseratAngebot.asp* die Tabelle, das heißt den darzustellenden Inhalt. Die Ergebnisse aus der Datenbankabfrage werden dort innerhalb einer *while*- Schleife verarbeitet, das heißt ausgelesen und (Reihe für Reihe) in die Tabelle eingefügt.

Danach hat der Nutzer die Möglichkeit, die angezeigten Tabellen ein- bzw. auszuschalten, indem eine Checkbox entsprechend gesetzt wird. Die Variable `$zeigean` übergibt dabei den aktuellen Zustand (entweder *checked* oder *empty*). Unter Nutzung der Objektfunktion `DbPage()` bleibt dieser Zustand bei einem Selbstaufufr erhalten.

...

```
$page=new DbPage();  
$page->AddParam("zeigean", $zeigean);
```

...

4.3 Anmerkungen zur Verwendung von JavaScript

Wegen der teils problematischen Natur von Skripten sollte normalerweise auf deren Nutzung verzichtet werden, wenn andere Optionen existieren. Da bei den standardisierten Elementen an einigen Stellen JavaScript enthalten ist, wird für die korrekte Funktionalität des Dienstes jedoch die **Aktivierung von JavaScript** im Browser sowieso **empfohlen**. Deshalb wurde innerhalb der Programmierung dieses Dienstes aus praktischen Gründen teilweise auch darauf zurückgegriffen. Dies trifft hauptsächlich auf die Weiterleitung zu, welche sich auf diese Art während der Programmier- und Testphase als zuverlässiger erwiesen hat, und außerdem ergab sich so gleichzeitig eine bequeme Lösung für Wertübergaben und Fallunterscheidungen bei der Weiterleitung. Auf den Seiten des Dienstes, welche eine Eingangsprüfung bestimmter Parameter enthalten, wird für den Fall, dass ein fehlerhafter Aufruf vorliegt und JavaScript deaktiviert ist, per `noscript` ein Alternativaufruf eingeblendet (wegen nicht funktionierender Weiterleitung zur Startseite).

4.4 Startseite des Dienstes

- **Dateiname:**

- *index.asp*

- **Zweck der Seite:**

Die Startseite dient als Ausgangspunkt für die Nutzung des Dienstes. Alle Funktionsgruppen werden hier logisch gruppiert versammelt, und dem Nutzer übersichtlich zugänglich gemacht.

- **Aufruf bzw. übergebene Parameter:**

- (1) Anwahl des Dienstes aus dem Studentenportal; Aufruf von den jeweiligen Dienstseiten teilweise durch Abbruch der Eingabe bei den Formularen.

- Parameter:

- In diesem Fall werden keine konkreten Parameter übergeben. Die einzige beim Aufruf benötigte Variable ist der angemeldete Nutzer, welcher durch die erforderliche Intranetanmeldung über die globale Variable `$REMOTE_USER` zur Verfügung steht.

- (2) Weiterleitung durch fehlerhaften Aufruf einer anderen Seite

- Parameter:

- Wird eine Seite des Dienstes aufgerufen, die entweder eine Eintragung vornehmen (Inserate, Bewertung, etc.) oder bereits bestehende Informationen anzeigen soll (Detailseite der Inserate etc.), dann wird der konkrete Aufruf hinsichtlich diverser kritischer Parameter auf Korrektheit geprüft. Das heißt, darf bspw. der Nutzer den Eintrag überhaupt aufrufen bzw. bearbeiten, oder wurden die Parameter manipuliert und existiert der damit angeforderte Eintrag eigentlich?

- `$message_error_aufruf` → Dieser Parameter enthält die einzublendende Nachricht, welche den Grund des fehlerhaften Aufrufs mitteilt.

▪ **Screenshot:**


Mitfahrerservice

Hochschule Mittweida (FH)
University of Applied Sciences

[Login: jbedlews/](#)
[\[Nach oben \]](#)
[\[Detailsuche \]](#)
[\[persönliches Profil bearbeiten \]](#)
[\[Kfz - Profil bearbeiten \]](#)
[\[detaillierte Personeninfos \]](#)
[\[Angebot / Gesuch eintragen bzw. ändern \]](#)
[\[Angebote/Gesuche Übersicht \]](#)
[\[Buchungsübersicht \]](#)
[\[Bewertungsübersicht \]](#)

angemeldeter Nutzer: Bedlewski, Jana

Mittwoch, der 24. November 2010, 13:50 Uhr



Webdienst zur Anbahnung von Mitfahrgelegenheiten <mit³>:

- Angebote und Gesuche können eingetragen/geändert/gelöscht werden.
- Eine Suchfunktion erlaubt Recherchen in allen Angeboten/Gesuchen.
- Die Buchungsfunktion dient zur Reservierung von freien Plätzen.
- Eine Bewertung für Fahrer und Mitfahrer ist möglich.
- Ein Nutzerprofil ermöglicht den Zugriff auf personenbezogene Details.

Haftungsausschluss: Es wird keine Haftung dafür übernommen, dass Fahrten tatsächlich stattfinden und getroffene Vereinbarungen eingehalten werden. Es kann kein rechtlicher Anspruch gegenüber dem Betreiber des Dienstes geltend gemacht werden.

<p>Angebote/Gesuche finden & buchen</p> <div> <input type="text" value="Start-/Zielort"/> <input type="text" value="Datum"/> <input type="button" value="Suchen"/> </div> <p> 6 Angebot(e) vorhanden 0 Gesuch(e) vorhanden </p> <p> Detailsuche (Suche mit erweiterten, teils optionalen Suchkriterien) </p>	<p>Suchen von Fahrern bzw. Mitfahrern in vorhandenen Angeboten/Gesuchen. Buchen von passenden Angeboten/Gesuchen.</p>
<p>Angebote/Gesuche erstellen & verwalten</p> <ul style="list-style-type: none"> Angebote/Gesuche eintragen Angebote/Gesuche Übersicht <ul style="list-style-type: none"> 3 Angebot(e) eingetragen 0 Gesuch(e) eingetragen 0 Fahrt(en) vorgemerkt 	<p>Erstellen und Verwalten, Bearbeiten und Löschen von Angeboten oder Gesuchen zum Mitfahren.</p>
<p>Buchungen verwalten</p> <ul style="list-style-type: none"> Buchungsübersicht <ul style="list-style-type: none"> 1 Angebot(e) von ihnen gebucht 0 Gesuch(e) von ihnen gebucht 2 Angebot(e) bei ihnen gebucht 0 Gesuch(e) bei ihnen gebucht 	<p>Verwalten aller Buchungen und Bearbeiten/Löschen der eigenen Buchungen.</p>
<p>Bewertungen abgeben & verwalten</p> <ul style="list-style-type: none"> Bewertungsübersicht <ul style="list-style-type: none"> 51 Bewertung(en) abgeschlossen 6 Bewertung(en) freigegeben 	<p>Eigene Bewertungen abgeben und alle von ihnen bzw. für Sie abgegebenen Bewertungen ansehen.</p>
<p>Profile</p> <ul style="list-style-type: none"> Profile eintragen: <ul style="list-style-type: none"> persönliches Profil (Pflicht) Fahrzeugprofil (optional) Profile ansehen: <ul style="list-style-type: none"> eigenes Profil <div> <input type="text" value="(Vor)Name/Sem.Gr."/> <input type="button" value="Suchen"/> </div>	<p>persönliche Daten eintragen und nach anderen eingetragenen Personen suchen.</p>

Rechtshinweis:
 Informationen zur Rechtslage bei Fahrgemeinschaften können Sie [hier](#) auf der Seite des ADAC erhalten.

Abbildung 4-1 Screenshot Startseite

▪ **Funktionsweise:**

(1) Anwahl aus dem Studentenportal / den jeweiligen Dienstseiten

Beim Aufruf des Mitfahrdienstes über das Studentenportal gelangt der Nutzer direkt zur Startseite. Als erstes werden einige Statusprüfungen durchgeführt, um den Dienst in aktuell angepasster Form zu präsentieren. Abgelaufene einmalige Inserate werden blockiert (sind damit über die Suche nicht mehr zugänglich) oder direkt gelöscht, wenn keine Buchung vorliegt. Passend dazu werden die zugehörigen Eintragungen in den persönlichen Favoritenlisten der Nutzer entfernt. Weiterhin werden die blockierten Bewertungen regelmäßiger

Fahrten falls möglich wieder freigegeben. Darüber hinaus wird durch den vorliegenden Login geprüft, ob der Nutzer bereits sein persönliches Profil vervollständigt hat, welches obligatorisch für die vollständige Nutzung des Dienstes ist. Anhand des Logins werden außerdem die auf den Nutzer angepassten Werte für die Anzahl der eingetragenen, gebuchten und vorgemerkten Angebote und Gesuche, sowie der abgeschlossenen und freigegebenen Bewertungen ermittelt und entsprechend angezeigt. Außerdem kann direkt über die Startseite eine Suche nach Inseraten und Profilen beteiligter Nutzer vorgenommen werden. Hierbei handelt es sich eine vereinfachte Suchmöglichkeit. Angebote und Gesuche können so durch Angabe eines Ortes und/oder des Datums und Profile über den Vor-/Nachname eines Nutzers oder dessen Seminargruppe gefiltert werden. Die Eingabefelder sind dabei mit einer grauen Beschriftung (= Legende) vorbelegt, welche jedoch verschwindet sobald der Nutzer in das betreffende Feld klickt (die Schriftfarbe ändert sich gleichzeitig auf das standardmäßige schwarz).

(2) Weiterleitung durch fehlerhaften Aufruf einer anderen Seite

Wenn eine Seite des Dienstes aufgerufen wird, bei welcher bestimmte Parameter korrekt vorliegen müssen, dann werden diese auf ihre Gültigkeit geprüft. Liegt hierbei ein kritischer Fehler vor, erfolgt die Weiterleitung zur Startseite inklusive Übergabe des Fehlertextes. Dieser wird entsprechend rot unterlegt angezeigt.

4.5 Seite zur Erstellung eines Angebots / Gesuchs

- **Dateiname:**

- *formular.MfzInserat.asp / MfzInserat.asp /
MfzInseratUpdateAchtung.asp*

- **Zweck der Seite:**

Über diese Seite hat der Nutzer die Möglichkeit, eigene Angebote oder Gesuche in die Datenbank einzutragen. Weiterhin wird dieses Formular dazu genutzt, bestehende Eintragungen zu modifizieren, zu kopieren oder Rückfahrten einzutragen.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

→ Parameter:

Bei diesem Aufruf werden keine wesentlichen Parameter übergeben. Das Formular wird lediglich initialisiert und ist bereit für die Eingabe eines neuen Inserats. Nur die Variable `$back` wird bei allen Seiten (außer der Startseite) übergeben, zu denen das Formular bei Abbruch der Eingabe weiterleiten bzw. zurückgehen soll (die Startseite ist als *default* Wert definiert und muss nicht extra übergeben werden). Handelt es sich dabei um eine Seite, bei welcher zwischen verschiedenen Nutzern unterschieden werden muss (siehe Inserats- und Bewertungsübersicht), dann wird auch die Variable `$user_ref` entsprechend weitergeleitet.

(2) Selbstaufruf der Seite nach manueller Eingabeprüfung

→ Parameter:

Bei jedem POST Vorgang des Formulars, zu dem auch die Prüfung gehört, werden diverse Parameter *hidden* übergeben. Überwiegend handelt es sich dabei um kritische Inseratsdetails, welche erst bei den schon bestehenden Inseraten zur Geltung kommen, und dort hauptsächlich für die indirekte bzw. weitergehende Eingabeprüfung benötigt werden. Die eigentliche Check-Funktion führt die standardmäßige Objektprüfung durch, und ist damit ausschließlich für die direkte Prüfung des Formulars selbst auf korrekte Eingabe zuständig.

(3) Aufruf der erstmaligen Eintragung der Rückfahrt durch Weiterleitung

→ Parameter:

Möchte der Nutzer eine Rückfahrt zu einer bereits vorgenommenen Eintragung erstellen und hat dies durch Auswahl des entsprechenden Radio-Buttons zum Ausdruck gebracht, wird nach dem Abspeichern des Haupteintrags das Inseratsformular neu aufgerufen und mit angepassten Parametern initialisiert (Startort Rückfahrt = Zielort Haupteintrag usw.). Um dies zu realisieren, werden folgende Parameter bei der Weiterleitung übergeben:

- `$ruckopt = aus` → damit wird die Rückfahrtoption ausgeblendet (d.h. die Radio-Buttons zur Auswahl der Rückfahrt) –

befindet sich der Nutzer in der Rückfahrt, ist dieser Parameter immer als „aus“ definiert und kann deshalb auch als Indikator genutzt werden

- `$ruckfirst = an` → damit wird die erstmalige Eintragung der Rückfahrt signalisiert
- `$hin_ref = $ref_iid` → hierbei handelt es sich um die Referenz- ID des Haupteintrages, welche benötigt wird, um die relevanten Details aus der Datenbank abzurufen und damit das Formular zu initialisieren. Bei der `$ref_iid` handelt es sich entweder um die letzte eingetragene ID (bei direkt aufeinander folgender Eingabe von Hin- und Rückfahrt), oder um die schon vorliegende feste ID der Hinfahrt (wenn Haupteintrag vorher separat vorgenommen wurde)
- `$message_ruck_ins = $message_ruck_ins` → dies ist die Hinweismeldung, um dem Nutzer die Rückfahrtseingabe anzuzeigen
- `$back = ag` → dieser Parameter wird benötigt, um beim Abbruch der Eingabe zur passenden Seite zurückzukehren, was in diesem Fall die Übersicht für Angebote/Gesuche ist

(4) Aufruf der bereits bestehenden zugehörigen Rückfahrt durch Weiterleitung (zur entsprechenden Modifizierung)

→ Parameter:

Existiert bereits eine Rückfahrt, wird die Weiterleitung nur bei Änderung des Haupteintrages ausgelöst. Der Rückfahrtsstatus wird dabei vorerst entzogen, für den Fall, dass die Rückfahrt letztendlich in den kritischen Parametern nicht mehr zur Hinfahrt passen sollte. Die Initialisierung des Formulars erfolgt mit den bereits vorgenommenen Angaben zur Rückfahrt (analog zum Formularaufruf bestehender Inserate). Dafür werden folgende Parameter bei der Weiterleitung übergeben:

- `$ruckopt = aus`
- `$hin_ref = $ref_iid`
- `$message_ruck_ins = $message_ruck_ins`
- `$back = ag`
- `$mitf_iid = $ruck_ref_inserat` → die ID der Rückfahrt wird zur Initialisierung des Inserats (als Objekt) übergeben
- `$updateopt = an` → zur Anzeige des möglichen Updates des bereits bestehenden Eintrags

(5) Aufruf eines bestehenden Eintrags zur Bearbeitung bzw. Kopie

→ Parameter:

Hierbei wird das Formular auf Basis eines bestehenden Eintrags initialisiert, entweder um diesen selbst zu bearbeiten, oder um eine Kopie davon zu erstellen. Folgende Parameter werden übergeben:

- `$mitf_iid` (Bearbeiten) / `$iid` (Kopieren) → bei der einfachen Bearbeitung wird die ID zur Initialisierung des Inserats (als Objekt) benötigt; beim Kopieren wird ein leeres Formular erzeugt, wobei mittels der ID die relevanten Parameter des Ursprungsinserts aus der Datenbank abgerufen und damit die entsprechenden Felder des neuen Eintrags vorbelegt werden
- `$opt = copy` (Kopieren) → zur Anzeige, dass es sich hierbei um eine Kopie handelt
- `$updateopt = an` (Bearbeiten) → zur Anzeige des möglichen Updates des bereits bestehenden Eintrags
- `$back = ag`

(A) MfzInseratUpdateAchtung.asp

→ Parameter:

Diese Seite wird ausschließlich über das Formular für Angebote bzw. Gesuche bei Bedarf aufgerufen – entweder bei der Speicherblockierung durch vorliegende Buchungen oder durch unzulässige Änderungen bei Rückfahrten.

- `$mitf_iid = $mitf_iid` → Die ID wird bei bestehenden Einträgen zum Abruf zusätzlicher Informationen benötigt.
- `$message_achtung_ruck = $message_achtung_ruck / $message_achtung = $message_achtung` → Diese Warnungsmeldung wird entsprechend den betroffenen Parametern erstellt. Damit soll der Nutzer direkt und deutlich auf die problematischen Punkte aufmerksam gemacht werden.
- `$ruckfirst = $ruckfirst` → Bei einer inkorrekten Rückfahrtseintragung wird mit Hilfe dieses Parameters unterschieden, ob im Falle der endgültigen Abspeicherung der ein Update eines bestehenden Eintrags stattfindet, oder (bei der erstmaligen Eintragung der Rückfahrt) ein neuer Inseratseintrag erstellt wird.
- `$hin_update / $ruck_update` → `$hin_update` wird für eine eventuelle Weiterleitung zur Rückfahrt benötigt. Die zweite Variable dagegen ist verantwortlich für die Unterscheidung bei der Anzeige der jeweiligen Hinweismeldungen und des zugehörigen Buttons. Bei vorliegenden Buchungen werden Hin- und Rückfahrt identisch behandelt - `$ruck_update` ist nur relevant bei einem (kritischen) Update einer ungebuchten Rückfahrt.
- `$startort / $startort_achtung(Buchungen) ... etc.` → Hierbei handelt es sich um die Parameter in ihrer aktuellen, aber (noch) nicht abgespeicherten Form. Für die Generierung der Benachrichtigungsmail, oder die endgültige Abspeicherung bei Rückfahrten muss diese Übergabe stattfinden.

- `$count_buch (Buchungen)` → Die Anzahl der bisherigen Buchungen wird als ein Detail für die Anpassung der Benachrichtigungsmail verwendet.

▪ **Screenshot:**

Angebot / Gesuch eintragen bzw. ändern Hochschule Mittweida (FH) University of Applied Sciences

Login: jbedlews/ [Nach oben]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:01 Uhr

Start Meine Angebote/Gesuche Meine Buchungen Meine Bewertungen Kontaktinfos

Angebot / Gesuch eintragen

Art Art: ☐ Angebot ☐ Gesuch

Zeit Frequenz: einmalig Datum: Uhrzeit: ☐ VHS ☒ Uhr

Strecke Startort: über: über: über: Zielort:

Details Treffpunkt: Plätze: 2 ☒ Preis pro Platz: ☐ VHS € Rückfahrt?: ☐ ja ☐ nein

Zusatzinfos Bemerkungen:

Abbrechen Prüfen Speichern

☒ korrekte Eingabe prüfen
☐ Datumsauswahl per Kalender
 'VHS' - Verhandlungssache
 rot - Pflichtangabe
 blau - freiwillige Angabe

Abbildung 4-2 Screenshot Inseratsformular

▪ **Funktionsweise:**

(1) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

Das Formular wird leer initialisiert, und ohne übergebene ID (`$mitf_iid =` Primärschlüssel der Inseratstabelle) wird somit jede abgeschlossene Eintragung als neuer Datensatz in der Tabelle für Angebote bzw. Gesuche eingefügt. Als Teil der **Eingangsprüfung** wird abgeklärt, ob der Nutzer sein persönliches Profil vervollständigt hat - dieses darf sich außerdem nicht im inaktiven Zustand befinden (`$status_profil / $count_user`). Sollte eine dieser Kriterien nicht zutreffen, währe der Eintrag ungültig, und die Weiterleitung zur Startseite wird direkt beim Aufruf ausgelöst.

Wurden auch beim Fahrzeugprofil Angaben gemacht, und ist dort eine Platzanzahl festgelegt, dann wird diese Zahl hier übernommen und in das entsprechende Feld eingetragen.

Entschließt sich der Nutzer das Inserat abzuspeichern, wird nach dem Betätigen des entsprechenden Buttons automatisch zunächst die Eingabeprüfung durchgeführt. Diese teilt sich in zwei Aspekte: die objektbasierte „Standardprüfung“ der Formularelemente auf Pflichteingaben bzw. korrekte Eingabe nach vorgegebenem Objektschema und die individuell ergänzten Prüfungskriterien. Teil eins wird bei jedem Prüfungsvorgang, das heißt auch bei erstmaliger Eingabe, durchlaufen. Die weitergehenden Prüfungen beinhalten auch Abschnitte, welche nur für bestehende Inserate und eventuell vorliegende Buchungen und/oder Rückfahrten relevant sind. Bei der hier vorliegenden Nutzungssituation sind folgende **allgemeine Prüfungskriterien** (und automatische Korrekturen) von Belang, welche auch für alle weiteren Verwendungen dieses Formulars Gültigkeit besitzen:

- Vollständigkeit der Pflichtangaben (rot gekennzeichnet)
- korrektes Datumsschema
- liegt Datum in der Zukunft?
- passt Datum zur gewählten Frequenz (gilt nur für regelmäßige Wochentage)?
- Korrektes Schema der Uhrzeitangabe
- Minimal-/Maximalanzahl von Sitzplätzen
- Prüfung auf Doppeleintragung – falls der Nutzer bspw. nach dem Speichern im Browser zurück geht und erneut abspeichert, würde anstatt einem Update ohne vorliegende ID der selbe Eintrag erneut erfasst werden (was mit dieser Prüfung verhindert werden soll)
- automatische Großschreibung der Orte

Sollte die Eingabe fehlerhaft sein, wird der Nutzer durch Einblendung von entsprechenden Fehlermeldungen darauf aufmerksam gemacht. In diesem Fall erfolgt ein Selbstaufzuruf der Seite, da eine Speicherung umgangen wird und der Nutzer so die Möglichkeit hat, entsprechende Änderungen vorzunehmen. Bei korrekter Eingabe erfolgt die Weiterleitung zur Übersicht für Angebote und Gesuche.

(2) Selbstaufzuruf der Seite nach manueller Eingabeprüfung

Bei fehlerfreier Eingabe wird der Nutzer nach dem Speichern auf eine andere Seite weitergeleitet. Deshalb wird der Selbstaufzuruf bei diesem Formular nur ausgelöst, wenn die separate Eingabeprüfung betätigt wird. Diese stellt ein Absenden (post) des Formulars dar, weshalb weitergehende (individuell erstellte) Prüfungsmechanismen auf unkomplizierte Weise implementiert werden können, indem diese einfach nach dem `$post` platziert werden. Die standardisierte Formularprüfung wird mittels der Funktion `check()` realisiert (`$obj->check()`).

(3) Aufruf der erstmaligen Eintragung der Rückfahrt durch Weiterleitung

Grundlegend wird in diesem Fall das Formular für einen neuen Eintrag initialisiert, aber mit dem Unterschied dass der Großteil der Formularelemente für die Rückfahrt passend vorbelegt wird. Würde man einfach die ID der Hinfahrt

zur Initialisierung nutzen - so müssten die ganzen einzelnen Zuweisungen nicht gemacht werden - dann würde der Nutzer ja nur den Hinfahrtseintrag updaten, anstatt ein neues Inserat zu erstellen. Die Vorbelegung befindet sich in der Definitionsdatei und besteht aus der Datenbankabfrage (unter Nutzung von `$hin_ref` als Referenz der Hinfahrt), und der Zuweisung der benötigten Daten zu den betreffenden Formularelementen. Hier als Beispiel Start- und Zielort, welche bei der Rückfahrt vertauscht werden müssen:

```
...  
  
$start_ref = $row_ref["startort"];  
$ziel_ref = $row_ref["zielort"];  
$this->startort=$ziel_ref;  
$this->zielort=$start_ref;  
  
...
```

Im Rahmen der Kompatibilität zur Hinfahrt sind alle Angaben noch modifizierbar, und die Speicherung muss wie bei jeder Eintragung noch bestätigt werden. Der positive Rückfahrtsstatus selbst wird erst endgültig gesetzt, wenn die Speicherung erfolgt ist. Damit kann der Nutzer die Eingabe noch problemlos abbrechen.

Sind jedoch kritische Änderungen angedacht (siehe Seite 43), erfolgt eine Weiterleitung zur Seite *MfzInseratUpdateAchtung.asp*, anstatt der Speicherung selbst. Dort hat der Nutzer noch die Möglichkeit zu vorherigen Seite zurückzukehren und entsprechende Anpassungen vorzunehmen, oder die erwünschten Modifikationen endgültig zu bestätigen und damit abzuspeichern.

Die Prüfung der Parameter selbst erfolgt nach dem POST des Haupteintrags, wobei die auf Seite 43 erwähnten Kriterien getestet werden. Art des Inserats und sämtliche Orte werden einfach nur mit dem Hinfahrtseintrag verglichen und bei jeglicher Abweichung wird der entsprechende Indikator aktiviert:

```
...  
  
if (($startort != $ziel_hin)){ $ruck_change_start = "an"; }  
  
...
```

Beim Datum wird zusätzlich sichergestellt, dass die Rückfahrt nicht vor der Hinfahrt liegt.

Die Frequenzprüfung ist etwas aufwendiger, da hierbei diverse Fallunterscheidungen nötig sind (je nach festgelegter Frequenz sind verschiedene Änderungen erlaubt bzw. unzulässig). Als Beispiel hier ein Teil der Prüfung für „werktags“ als Frequenz der Hinfahrt:

```
...  
  
if ($frequenz_hin == "werktags")  
{  
    if ($auswahl_frequenz == "einmalig"){ $nochange_freq =  
        "an"; }  
    if ($auswahl_frequenz == ("montags" || "dienstags" ||  
        "mittwochs" || "donnerstags" || "freitags" ||  
        "samstags" || "sonntags")){ $nochange_freq = "an"; }  
}  
  
...  
  
($auswahl_frequenz = aktuell gewählte Frequenz)
```

(4) Aufruf der bereits bestehenden zugehörigen Rückfahrt durch Weiterleitung

Hin- und Rückfahrt müssen bei einigen kritischen Details übereinstimmen bzw. zusammenpassen, um eine sinnvolle Kombination zu ergeben. Werden an der Hinfahrt Änderungen vorgenommen, sollte die Rückfahrt (bei Bedarf) dementsprechend angepasst werden. Dabei werden alle Parameter nach dem *post*- Vorgang mit ihrer vorherigen Variante verglichen, und bei jeglichen Unterschieden wird der Indikator `$hin_update = "an"` aktiviert.

Nach der Speicherung des Haupteintrags erfolgt die Weiterleitung zum entsprechenden Eintrag der Rückfahrt. Die ID der Rückfahrt wird übergeben, und das Inserat entsprechend initialisiert.

(5) Aufruf eines bestehenden Eintrags zur Bearbeitung bzw. Kopie

Im Grunde wird eine ID übergeben, und das Formular dementsprechend initialisiert (bestehendes Inserat) bzw. vorbelegt (Kopie). Beim Aufruf des Formulars wird zunächst jedoch erst einmal geprüft, ob der angeforderte Eintrag überhaupt existiert, dem betreffenden Nutzer gehört und noch gültig ist (Status!=1).

Bei der Kopie werden im Gegensatz zur ersten Eintragung der Rückfahrt alle im Formular enthaltenen Details exakt nach Vorgabe des entsprechenden Inserats zugewiesen.

Beispielsweise:

```
...  
  
$start_copy = $row_copy["startort"];  
$this->startort=$start_copy;  
  
...
```

Änderungen bestehender Inserate werden (mit Ausnahme einer möglichen Weiterleitung zur Rückfahrt) über einen solchen Aufruf aus der Übersicht für Angebote und Gesuche realisiert. Dabei gibt es Einschränkungen was die

Modifikation von Einträgen betrifft, bei denen schon mindestens eine Buchung vorliegt (siehe Seite 42).

Nach dem *post*- Vorgang wird geprüft, ob für das betreffende Inserat bereits Buchungen vorhanden sind. Danach werden alle Einträge mittels einer *while*-Schleife durchlaufen...

...

```
while($row_buch_check=mysql_fetch_array($result_buch_check))  
{
```

...

... und die jeweiligen relevanten Buchungsdetails werden ausgelesen. Das bedeutet, für jeden Parameter wird einzeln geprüft, ob und (wenn ja) in welcher Form dieser in der betreffenden Buchung vorhanden ist bzw. sich darstellt. Dabei werden diverse Vergleiche und *switch/case*- Anweisungen durchlaufen. Bei Übereinstimmung wird der passende Indikator um eins erhöht. Hier als Beispiel die Prüfung, ob der erste Zwischenhalt als Start- oder Zielort gebucht wurde.

...

```
if (($start_buch == $ueber1_inserat) || ($ziel_buch ==  
$ueber1_inserat))  
{  
    if (!$count_ueber1){ $count_ueber1 = 1; }else{  
        $count_ueber1 = $count_ueber1 + 1;}  
}
```

...

Die Frequenzprüfung ist aufgrund der vielfältigen Optionen etwas umfangreicher. Sowohl die möglichen Frequenzen an sich, als auch die einzelnen Wochentage (bei einmaligen und Auswahlbuchungen) werden ermittelt.

Bei der Modifikation der Sitzplatzanzahl darf die neu gewählte Menge nicht unterhalb der bereits gebuchten liegen. Deshalb wird die Minimalzahl der benötigten Plätze ermittelt:

...

```
if ($buch_mon_reg > $buch_die_reg){ $pl_zw = $buch_mon_reg;  
}else{ $pl_zw = $buch_die_reg; }
```

...

(hier abgebildete Parameter sind „bereinigt“, d.h. Einzelbuchungen wurden abgezogen → regelmäßige und einmalige Fahrten können sich bedingt überschneiden und würden so die Zahl verfälschen)

Ist die aktuell angegebene Anzahl gültig, müssen noch die einzelnen Wochentage unter Abzug bisheriger Buchungen aktualisiert werden.

Der Indikatoren werden letztendlich genutzt, um (bei ungültigen Änderungen) entsprechend die passende Mitteilung an den Nutzer zu generieren, die

direkte Speicherung zu sperren und die Weiterleitung zur Hinweisseite *MfzInseratUpdateAchtung.asp* auszulösen.

Sollten jedoch nur kleinere Modifikationen angedacht sein, ist dies jederzeit möglich. In dem Fall wird an die buchenden Personen eine Nachricht per Mail geschickt, um auf die geänderten Parameter hinzuweisen.

Wird ein Inserat aufgerufen, welches bereits eine Rückfahrt besitzt, kann diese Option auch wieder zurückgezogen werden. Setzt der Nutzer den entsprechenden Radio-Button auf „nein“, wird die Rückfahrt automatisch gelöscht. Dies trifft auch zu, wenn bereits Buchungen vorhanden sind (Löschen einer Fahrt muss immer möglich sein). Dabei wird eine entsprechende Benachrichtigungsmail an die betreffenden Personen geschickt:

```
...  
  
while ($row_buch=mysql_fetch_array($result_buch))  
{  
  
    ...  
  
    $id = $row_buch[pid];  
  
    mail_inserat_cancel($id, $art_ruck, $start_ruck,  
        $ziel_ruck, $dat_ruck);  
}  
  
...
```

Analog zum eigenständigen Löschen von Inseraten, werden auch in diesem Fall die zugehörigen Einträge der Buchungen, des (temporären) Bewertungsvorganges, der Favoriten und des Inserates selbst entsprechend behandelt. Das heißt, hat die Fahrt noch nicht stattgefunden, werden alle Einträge gelöscht. Ansonsten werden Buchungs- und Inseratseintrag inaktiv gesetzt (Status = 1), der Verweis auf die Rückfahrt gelöscht, und (für den Fall das Hin- und Rückfahrt gebucht wurden) bei Bedarf der korrespondierende temporäre Bewertungseintrag freigegeben.

(A) *MfzInseratUpdateAchtung.asp*

Diese Seite ist für zwei verschiedene Situationen zuständig:

Den Nutzer darauf aufmerksam zu machen...

- ... dass aufgrund vorliegender Buchungen die erwünschten Änderungen nicht vorgenommen werden können (es besteht die Möglichkeit eine Hinweismail an alle Buchenden zu senden)
- ... dass (bei Rückfahrten) kritische Änderungen noch nicht endgültig sind und damit entweder noch verworfen oder auf dieser Seite abgeschlossen werden können (bei Verlust des Rückfahrtsstatus)

(Die wesentliche Unterscheidung beider Fälle wird nach dem *post* des Formulars getroffen.) Je nach Aufruf wird entweder der „Absenden“ Button bei Buchungen oder der „Speichern“ Button bei Rückfahrten eingeblendet. Durch diese Unterscheidung werden nach dem *post*- Vorgang des Formulars die jeweils entsprechenden Aktionen durchgeführt.

Innerhalb der Schleife...

...

```
if ($updateruck_x) {
```

...

... wird die endgültige Speicherung der veranlassten Änderungen bearbeitet. Zuerst wird entsprechend der gewählten Frequenz die Platzanzahl auf die passenden Wochentage zugewiesen. Bei einem Update eines bestehenden Eintrages werden alle Parameter entsprechend der vorangegangenen Eingabe aktualisiert. Außerdem werden der Rückfahrtsstaus entzogen, sowie die zugehörigen temporären Bewertungseinträge freigegeben. Handelt es sich um die erstmalige Eintragung einer Rückfahrt, dann erfolgt an dieser Stelle der eigentliche Speichervorgang. Abschließend erfolgt die Weiterleitung zur Übersicht für Angebote und Gesuche.

Die Schleife...

...

```
if ($updatemessage_x) {
```

...

... ist allem voran für die Bearbeitung der Hinweismails zuständig (an die Personen die das betreffende Inserat gebucht haben). Falls es sich bei dem aktuellen Eintrag um ein Inserat mit zugehöriger Rückfahrt handelt, wird auch die Weiterleitung dahin veranlasst (analog zur Modifikation einer ungebuchten Fahrt).

4.6 Seiten für die Übersicht eingetragener Angebote / Gesuche

- **Dateiname:**

- *MfzInseratUebersicht.asp*
- *MfzInseratAngebot.asp*
- *MfzInseratGesuch.asp*
- *MfzInseratFav.asp*

- **Zweck der Seite:**

Auf dieser Seite hat der Nutzer alle seine selbst eingetragenen Inserate im Blick, und kann diese auf verschiedene Art managen. Detailansicht, Bearbeiten, Kopieren und Löschen sind die angebotenen Optionen. Es gibt außerdem die Möglichkeit, eingetragene Inserate anderer Nutzer vorzumerken, ohne diese gleich buchen zu müssen.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- Da es sich hierbei nicht um ein Formular handelt, werden keine Parameter *hidden* übergeben. Jedoch werden immer ein paar Werte für die ordnungsgemäße Darstellung benötigt, weshalb hier zu diesem Zweck zwei verschiedene Hilfsmöglichkeiten kombiniert werden. Die Verwendung einer Session und der Objektfunktion `DbPage()` sichern den Erhalt der Variablen im Falle eines Selbstaufrufs.

- `$id, $sid` → Die `$sid` ist der verwendete Name der Session. Wird ein Login übergeben (siehe Profilansicht), hat sich gezeigt, dass auch die Session-ID mit übergeben werden sollte (um Konflikte zu vermeiden wird die Variable noch entsprechend auf `$sid` zugewiesen)
 - `$pid_user` → Diese Variable wird für die konkrete Datenbankabfrage bezüglich der eingetragenen Inserate verwendet und dient als einer der beiden Richtwerte der angepassten Darstellung der Übersicht. Der (ggf. übergebene) Nutzer Login wird auf diese Variable abgebildet.
 - `$showref` → Über diese Variable wird gesteuert, welche der Tabellen angezeigt wird (Angebote, Gesuche oder alle). Der Parameter `$ref` wird auf diesen Wert übergeben.
 - `$sortan, $orderan, $richtungen` → Diese Variablen steuern die Sortierung der jeweiligen Tabellen, in diesem Fall die der Angebote. `$sortan` wird als Parameter im Link übergeben, den der Nutzer zum Sortieren anklickt. `$orderan` ist die Kombination aus der entsprechenden Tabelle in der Datenbank und dem angeklickten Suchparameter (diese Variable wird für die

Abfrage benötigt). `$richtungan` bestimmt, ob absteigend oder aufsteigend sortiert wird.

- `$sortge`, `$orderge`, `$richtungge` → Sortierung der Gesuche
- `$sortfav`, `$orderfav`, `$richtungfav` → Sortierung der vorgemerkten Inserate
- `$zeigean`, `$zeigege`, `$zeigefav` → Bei den Tabellen ist es möglich, diese durch Aktivierung einer Art Checkbox an- und auszuschalten, bzw. zu minimieren. Über diese Variablen wird der momentane Status gespeichert.

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten (`$REMOTE_USER`)

→ Parameter:

Ruft der angemeldete Nutzer die eigene Übersicht auf, wird kein Parameter übergeben, da der `$REMOTE_USER` als default – Wert definiert ist, und diese globale Variable immer zur Verfügung steht.

(3) Aufruf über die Profilansicht (andere Nutzer/`$REMOTE_USER`)

→ Parameter:

Über die Profilansicht eines Nutzers können auch dessen aktuelle Angebote und Gesuche direkt aufgerufen werden, ohne erst über die Suche gehen zu müssen. Da dieser Aufruf auch für andere Nutzer zugänglich ist, muss die ID der betreffenden Person übergeben werden, weil nicht alle Aspekte dieser Übersicht für fremde Nutzer bestimmt sind und die Anzeige entsprechend angepasst werden muss.

- `$user1 = $pid_user` → Hierbei handelt es sich um den an die Profilseite übergebenen Login
- `$ref` → Angebote und Gesuche sind über separate Links zugänglich, und werden über diesen Parameter unterschieden
- `$zeigean = empty` / `$zeigege = empty` → Es können entweder die Angebote oder die Gesuche eines anderen Nutzers ausgewählt werden. Zur korrekten Anzeige wird die jeweils andere Tabelle ausgeblendet.
- `$id = md5 (uniqid (rand()))` → In der Übersicht ist es möglich, die Tabellen nach diversen Kriterien zu sortieren. Dabei findet ein Selbstaufruf der Seite statt. Damit die nutzerbezogene Darstellung korrekt erhalten bleibt, wird in der Übersicht eine Session verwendet, wobei die übergebene ID der Name der Session ist. In diesem Fall wird die ID zufällig generiert, was sich im Gegensatz zu einem fest voreingestellten Wert beim Testen als zuverlässiger erwiesen hat.

▪ **Screenshot:**

Angebote/Gesuche Übersicht

Hochschule Mittweida (FH) University of Applied Sciences

Login: jbedlews/ [Nach oben]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:03 Uhr

[Start](#)
[Neues Angebot/Gesuch](#)
[Meine Buchungen](#)
[Meine Bewertungen](#)
[Kontaktinfos](#)

☒ 3 Angebot(e) eingetragen (J. Bedlewski)

Frequenz	Datum	Startort	Zielort	Buch.	
täglich	Di, 22. Jun 2010	Mittweida	Flöha	2	* f i x
montags	Mo, 20. Sep 2010	Flöha	Döbeln	0	* f i x
montags	Mo, 20. Sep 2010	Leipzig	Flöha	0	* f i x

☒ 1 Gesuch(e) eingetragen (J. Bedlewski)

Frequenz	Datum	Startort	Zielort	Buch.	
einmalig	Fr, 10. Dez 2010	Flöha	Unterföhring	0	* f i x

☐ 0 Fahrt(en) vorgemerkt (J. Bedlewski)

[Zurück](#)

* Angebot/Gesuch neu erstellen durch kopieren
 f Angebot/Gesuch bearbeiten
 i Details des Angebotes/Gesuches betrachten
 x Angebot/Gesuch (oder vorgemerkte Fahrt) löschen
 VHS - Verhandlungssache
 R - Rückfahrt (auch einzeln buchbar)

Abbildung 4-3 Screenshot Inseratsübersicht

▪ **Funktionsweise:**

(1) Selbstaufruf

Ein solcher Selbstaufruf wird bei zwei Situationen ausgelöst:

- dem Ein- bzw. Ausschalten der Tabellen über die Checkbox
- der Nutzung der Sortierungsfunktion (Anwahl der entsprechenden Links)

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten (\$REMOTE_USER)

Ohne übergebene ID (Login), wird der \$REMOTE_USER eingesetzt. Die ID der Session wird in diesem Fall auch erst direkt auf der Seite selbst zugewiesen.

Der Inserent hat bei den hier erwähnten Aufrufmöglichkeiten immer Zugriff auf alle Tabellen (Angebote, Gesuche und vorgemerkte Einträge) inklusive aller definierten Verwaltungsfunktionen.

Beim Aufruf der Seite (egal ob \$REMOTE_USER oder Fremdnutzer) wird wie bei allen nutzerbezogenen Seiten geprüft, ob das persönliche Profil eingetragen und aktiv ist.

(3) Aufruf über die Profilansicht (andere Nutzer/\$REMOTE_USER)

In der Profilansicht können entweder die Angebote oder die Gesuche eines Nutzers aufgerufen werden. Je nach getroffener Auswahl wird nur diese Tabelle angezeigt. Ruft der Nutzer seine eigenen Inserate auf, stehen ihm die Verwaltungsfunktionen auch hier zur Verfügung. Alle anderen Nutzer können nur die Details des Eintrags betrachten.

4.7 Seite zur Darstellung der Details der Angebote und Gesuche

- **Dateiname:**

- *MfzInseratDetails.asp / MfzInserat.asp*

- **Zweck der Seite:**

Auf dieser Seite werden alle Details eines bestimmten Angebotes oder Gesuches dargestellt.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf über die verschiedenen Übersichtsseiten/die Suchergebnisliste

→ Parameter:

- \$mitf_iid → Die ID des Eintrags für den Aufruf der Daten.
- \$buchopt = an (andere Nutzer) → Dieser Parameter wird benötigt, um den Buchungslink einzublenden.

- **Screenshot:**

The screenshot displays a web application interface for 'Hochschule Mittweida (HTW) University of Applied Sciences'. The user is logged in as 'jbedlews/'. The page shows details for an offer (Angebot) created by Sebastian Weiß on November 24, 2010. The offer is for a daily (täglich) negotiation case (Verhandlungssache) starting on December 1, 2010, at 08:00 Uhr. The location is an overpass (Oberführung) to an underpass (Unterführung). The meeting point (Treffpunkt) is Pharaohaus, with 2 seats (Plätze) available. The price per seat (Preis pro Platz) is 1 € for the negotiation case. The return trip (Rückfahrt) is not required. The page includes a 'Zurück' button at the bottom right.

Detailansicht	
Art	Insert von: Sebastian Weiß Art: Angebot eingetragen / geändert am: Mi, 24. Nov 2010
Zeit	Frequenz: täglich Datum: Mi, 1. Dez 2010 Uhrzeit: Verhandlungssache - Vorschlag: 08:00 Uhr
Strecke	Startort: Oberführung über: (keine Angabe) Zielort: Unterführung
Details	Treffpunkt: Pharaohaus Plätze: 2 aktuelle Buchungen: 0 Preis pro Platz: Verhandlungssache - Vorschlag: 1 € Rückfahrt ?: nein
Zusatzinfos	Bemerkungen: (keine Angabe)
Zusatzoptionen	Plätze buchen ?: <input type="checkbox"/> zum Buchen Fahrt merken ?: <input checked="" type="checkbox"/>

[Zurück](#)

Abbildung 4-4 Screenshot Inseratsdetails

- **Funktionsweise:**

- (1) Aufruf über die verschiedenen Übersichtsseiten/die Suchergebnisliste

Analog dem Formular zur Eintragung von Angeboten und Gesuchen, wird auch hier die übergebene ID für die Initialisierung der Inseratsdaten mittels eines Datenbankobjektes benötigt (für den Fall das die ID manipuliert wurde, wird geprüft ob ein passender Eintrag überhaupt existiert). Beide Seiten greifen auf dieselbe Definitionsdatei (*MfzInserat.asp*) zurück. Anstatt jedoch die einzelnen Parameter in die diversen Bearbeitungselemente zu laden, werden diese hier nur in Textform angezeigt.

...

```
$obj->startort
```

...

Alle Parameter, die nicht als Objekt definiert sind, werden über eine separate Datenbankabfrage in die Darstellung eingefügt.

Einige Parameter werden ausgelesen und in angepasster Form dargestellt. Etwa die Variablen, die nur als 1 oder 0 abgespeichert werden (die Checkboxen). Oder auch die Zwischenhaltepunkte, welche selbstverständlich nur bei vorhandener Eintragung angezeigt werden.

Als Zusatzfunktionen können alle Nutzer über diese Seite die persönlichen Details des Inserenten abrufen, oder bei vorhandener Rückfahrt den entsprechend korrespondierenden Eintrag direkt anklicken. An den Login angepasst werden jedoch die Anzeige der bisherigen Buchungen (nur für den Inserenten selbst bestimmt) und der Buchungsmöglichkeit (nur für die anderen Nutzer), sowie die Möglichkeit, den Eintrag vorzumerken.

4.8 Seite zum Löschen eingetragener Angebote und Gesuche

- **Dateiname:**

- *MfzInseratDelete.asp / MfzInserat.asp*

- **Zweck der Seite:**

Möchte der Nutzer einen seiner Einträge löschen, wird er auf diese Seite weitergeleitet. Hier kann er den Löschvorgang endgültig machen, oder sich noch dagegen entscheiden und den Vorgang abubrechen.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf von der Übersichtsseite

→ Parameter:

- \$mitf_iid → Hierbei handelt es sich um den Schlüssel des zu löschenden Eintrags.

- **Screenshot:**



Abbildung 4-5 Screenshot Löschen eines Inserats

- **Funktionsweise:**

(1) Aufruf von der Übersichtsseite

Einige wichtige Details zur Fahrt werden nochmals aufgelistet, und der Nutzer erhält die endgültige Aufforderung sich über die Ausführung des Vorgangs zu entscheiden. Beim Aufruf der Seite wird geprüft, ob für das betreffende Inserat bereits Buchungen vorliegen. Trifft dies zu, wird der Nutzer mit einem Warnhinweis darauf aufmerksam gemacht. Im Gegensatz zur Modifikation eines Angebotes oder Gesuches kann der Löschvorgang aber auch in diesem Fall durchgeführt werden. Die Zuverlässigkeit des Dienstes hat zwar Priorität, jedoch muss der Urheber des Inserats in letzter Konsequenz immer noch über seinen Eintrag entscheiden können.

Beim Löschen sind zwei Situationen zu unterscheiden:

- Die Fahrt wurde gebucht und ist bereits zur Bewertung freigegeben
- Die Fahrt ist ungebucht bzw. gebucht, hat aber noch nicht stattgefunden

Der Inseratseintrag steht im Zusammenhang mit vorliegenden Buchungen und den zugehörigen Bewertungseinträgen. Wird also eine bestimmte Fahrt gelöscht, müssen auch die verknüpften Datensätze entsprechend behandelt werden. **Im ersten Fall** ist ein direktes Löschen von Inserats- und Buchungseintrag nicht erwünscht, da ein Zugriff bei der späteren Bewertung noch möglich sein muss. Deshalb wird hier nur der Status auf 1 gesetzt, was den Eintrag deaktiviert. Erst wenn die nächste Bewertung erfolgt ist, werden die Datensätze dann endgültig gelöscht. **Fall zwei** erfordert diesen Zwischenschritt nicht, da keine Rücksicht auf die mögliche spätere Verwendung der Daten genommen werden muss. Dies gilt auch wenn bereits Buchungen vorliegen, da eine Fahrt ja sowieso erst bewertet werden kann, nachdem sie stattgefunden hat.

Unabhängig welcher Fall der Anwendung gerade vorliegt, müssen bei jedem „Löschvorgang“ die zugehörigen Favoriteneinträge aus der Datenbank entfernt, sowie (falls nötig) der Rückfahrtsstatus zurückgesetzt werden. Sollte dieses Inserat eine Hin- oder Rückfahrt besitzen und bei entsprechenden Buchungen ein zugehöriger temporärer Bewertungsdatensatz vorliegen welcher über `$bew_block_1` und `$bew_block_2` momentan gesperrt ist, dann wird dieser gelöscht (egal ob die Fahrt schon stattgefunden hat oder nicht). Das gilt auch für die zugehörige Buchung an sich, sowie das Inserat selbst (wenn keine weiteren Buchungen vorliegen). Da Hin- und Rückfahrt bewertungstechnisch als Einheit betrachtet werden, und in diesem Fall bereits ein Bewertungsdatensatz freigegeben ist, wird der Blockierte nicht mehr benötigt.

Auch beim Aufruf dieser Seite wird die Gültigkeit des Eintrags bezüglich der übergebenen ID geprüft – existiert eine solche Eintragung, ist diese noch gültig, gehört sie dem `$REMOTE_USER` und besitzt dieser einen aktiven Profileintrag?

4.9 Seite zum Löschen einer Eintragung in der Favoritenliste

- **Dateiname:**

- *MfzFavDelete.asp*

- **Zweck der Seite:**

Über diese Seite kann der Nutzer einen Eintrag in der Liste für vorgemerkte Fahrten wieder entfernen.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf von der Übersichtsseite für Angebote und Gesuche

→ Parameter:

- \$mitf_fid → Hierbei handelt es sich um die ID des zugehörigen Datenbankeintrages. Dieser Schlüssel reicht aus, um den Löschbefehl korrekt ausführen zu können.
- \$mitf_iid → Hierbei handelt es sich um die ID des zugehörigen Inserats. Diese wird für das Auslesen relevanter Inseratsdetails benötigt, und muss so nicht extra über einen zusätzlichen Datenbankabruf ausgelesen werden.

- **Screenshot:**

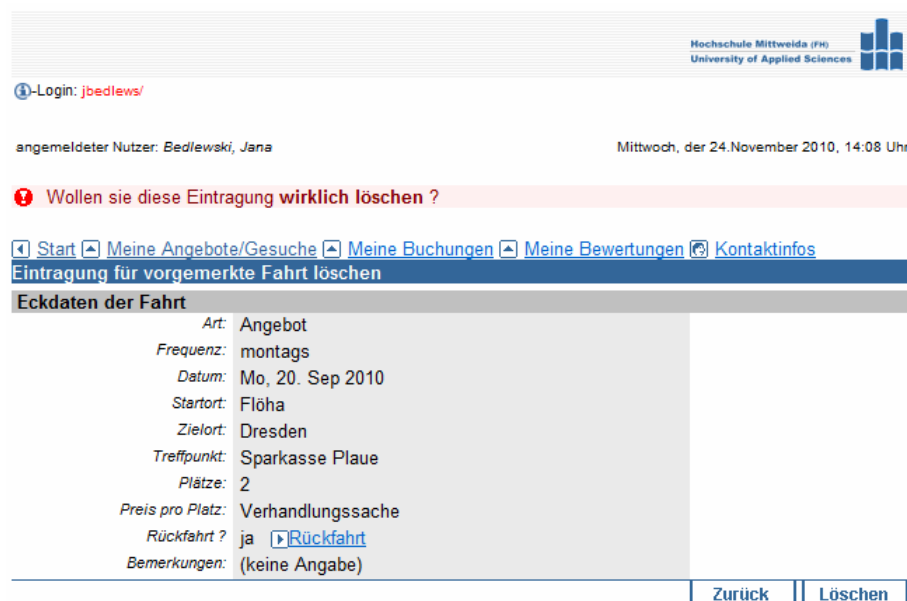


Abbildung 4-6 Screenshot Löschen eines Eintrags der Favoritenliste

- **Funktionsweise:**

(1) Aufruf von der Übersichtsseite für Angebote und Gesuche

Wie auch bei den anderen Seiten dieses Mitfahrdienstes, welche für das Löschen von Einträgen zuständig sind, werden hier nochmals wichtige Details der Fahrt aufgelistet, und der Nutzer wird gefragt, ob der Eintrag wirklich gelöscht werden soll. In diesem Fall ist die Verarbeitung eher simpel, da es sich hierbei eigentlich nur um einen einfachen Vermerk in der Datenbank handelt. Dieser ist außerdem nicht, wie etwa bei den Inseratseinträgen selbst, mit an-

deren Datensätzen verknüpft, weshalb dahingehend keine weitergehende Behandlung relevanter Parameter erforderlich ist.

Nachdem der Löschvorgang vom Nutzer bestätigt wurde, wird der Eintrag einfach per SQL-Befehl aus der Datenbank entfernt:

...

```
$sql_fav_delete="DELETE FROM mitf_fav WHERE  
mitf_fid='$mitf_fid';  
$result_fav_delete= db_query($sql_fav_delete);
```

...

Die obligatorische Eingangsprüfung auf Korrektheit der ID und des Nutzerprofils erfolgt auch bei dieser Seite.

4.10 Seiten zur Durchsuchung eingetragener Angebote / Gesuche

- **Dateiname:**

- *formular.DetSuche.asp / index.asp / MfzSuche.asp*

- **Zweck der Seite:**

Mittels der Suchfunktion ist es dem Nutzer möglich, Inseratseintragungen nach bestimmten Kriterien zu filtern und so nach eigenen Bedürfnissen passend aufzulisten. Im Gegensatz zu einer eher simplen Methode nach Art eines schwarzen Brettes oder eines Internetforums kann so wesentlich zielgerechter und mit größerer Übersichtlichkeit recherchiert werden, was die Nutzerfreundlichkeit erhöht und den Dienst so attraktiver macht. Dabei existiert eine einfache Suche, die sich auf Ort und Datum beschränkt und von der Startseite direkt nutzbar ist. Für die gründlichere Differenzierung der eingetragenen Angebote und Gesuche gibt es zusätzlich noch ein detailliertes Suchformular mit einer Vielzahl an Parametern zur genaueren Eingrenzung.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf (Detailsuche)

- Parameter:

- Ein Selbstaufruf ist bei diesem Formular so gar nicht vorgesehen, da keine direkte Eingabeprüfung existiert. Sollte jedoch ein *reload* der Seite erfolgen, würde das ja einen Selbstaufruf darstellen. Dabei werden aber keine Parameter übergeben.

- (2) Einfache Suche → direkt auf der Startseite

- Parameter:

- Da es sich hierbei um den Standardaufruf der Startseite handelt, werden keine speziellen Parameter für die Suche übergeben.

- (3) Detailsuche → Aufruf über die Startseite

- Parameter:

- Analog der Eintragung eines neuen Angebotes oder Gesuches, wird dieses Formular für eine neue Eingabe leer initialisiert. Das trifft auf jeden Aufruf zu, da bei diesem Verwendungszweck später kein Datensatz zwecks Modifikation zurück ins Formular geladen werden muss.

▪ **Screenshot:**

Abbildung 4-7 Screenshot Detailsuche

▪ **Funktionsweise:**

(1) Selbstaufruf (Detailsuche)

Falls es bei diesem Formular wegen Neuladens zu einem Selbstaufruf kommen sollte, ist zu beachten dass bei einem solchen Aufruf (genau wie beim Zurückgehen von der Ergebnisseite) das Formular wieder leer dargestellt wird (eventuelle vorherige Angaben sind hinfällig). Damit sollen mögliche Komplikationen mit den durch JavaScript gesteuerten Elementen vermieden werden (siehe „Ausschalten“ von Elementen durch Aktivierung einer Checkbox).

(2) Einfache Suche → direkt auf der Startseite

Bei dieser Suchoption für den schnellen Einstieg kann der Nutzer über die Parameter Ort und Datum Einfluss auf die anzuzeigende Ergebnisliste nehmen. Dabei ist zu beachten:

- es werden immer Angebote und Gesuche berücksichtigt
- Der Parameter „Ort“ bezieht sich auf Start, Ziel und mögliche Zwischenhalte der Fahrt
- Die Suche kann durchgeführt werden bei Eingabe von einem, gar keinem oder beiden Parametern in Kombination
- bei Suche ohne Angabe von Ort und/oder Ziel werden automatisch alle eingetragenen Inserate angezeigt

Wird der Suchvorgang über die Betätigung des Buttons bestätigt, erfolgt die Weiterleitung zur Ergebnisseite *MfzSucheErgebnisEinfachUebersicht.asp*, wo

die Angaben entsprechend verarbeitet und im Anschluss in einer Übersichtstabelle dargestellt werden.

(3) Detailsuche → Aufruf über die Startseite

Die Unterteilung des Formulars richtet sich nach dem zur Erstellung von Angeboten und Gesuchen, das heißt die Abschnitte Art, Zeit, Strecke und Details sind auch hier vorhanden. Ein inhaltlicher Unterschied ist jedoch in den Details zu finden, denn in der Suche können hierbei Angaben zur Person berücksichtigt werden. Wegen der Auslagerung auf das persönliche Profil muss dies bei der Eintragung nicht jedes Mal wiederholt werden.

Bei diesem Formular bieten sich dem Nutzer diverse Auswahlmöglichkeiten, um die Suche nach eigenen Vorstellungen anzupassen. Nicht benötigte Parameter die Fahrt selbst betreffend können über eine Checkbox als „egal“ definiert werden, was das betreffende Element per JavaScript ausschaltet. Nach demselben Prinzip ist es möglich, den Toleranzbereich bei Uhrzeit und Datum zu aktivieren, oder bestimmte persönliche Parameter einzubeziehen. So können auf Wunsch beispielsweise nur Einträge eines bestimmten Nutzers angezeigt werden. Dabei ist zu beachten, dass die Eingabe des Vor- und/oder Nachnamens, oder des Logins eines Nutzers akzeptiert wird.

Sollten Angaben offen gelassen worden sein, werden die betreffenden Parameter automatisch als „egal“ gesetzt, um die Suche trotzdem durchführen zu können.

Nach der getroffenen Auswahl erfolgt auch hier die Weiterleitung auf eine Ergebnisseite – *MfzSucheErgebnisUebersicht.asp* – zur Verarbeitung der Daten und Darstellung der Resultate.

4.11 Seiten zur Verarbeitung und Darstellung der Suchergebnisse

- **Dateiname:**

- *MfzSucheErgebnisUebersicht.asp*
- *MfzSucheErgebnisEinfachUebersicht.asp*
- *MfzSucheErgebnisAngebot.asp*
- *MfzSucheErgebnisGesuch.asp*

- **Zweck der Seite:**

Die über das Suchformular getroffenen Angaben werden auf dieser Seite verarbeitet und anschließend nach Vorbild der diversen im Mitfahrdienst verwendeten Übersichtsseiten in einer Tabelle getrennt nach Angeboten und Gesuchen aufgelistet.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- Zur Steuerung der Sortierung und korrekten Darstellung der Tabellen werden folgende Parameter übergeben:

- `$sortan`, `$orderan`, `$richtungen` → Steuerung der Sortierung der Tabelle für Angebote.
 - `$sortge`, `$orderge`, `$richtungge` → Steuerung der Sortierung der Tabelle für Gesuche.
 - `$zeige`, `$zeigege` → Zur Aufrechterhaltung des Aktivierungszustands über den Selbstaufruf hinweg, wird der Status in diesen Variablen gespeichert.

- Außerdem werden alle Suchparameter in einer Session gespeichert, um nach einem Selbstaufruf noch zur Verfügung zu stehen (zwecks erneuter Datenbankabfrage).

- (2) Aufruf durch Weiterleitung von der Startseite (einfache Suche)

- Parameter:

- Die Parameter des Suchvorgangs (hier nur Ort und Datum) werden von der als Formular definierten Suchmaske übergeben, und dann auf dieser Seite in der konkrete Datenbankabfrage verarbeitet.

- (3) Aufruf durch Weiterleitung vom Suchformular (Detailsuche)

- Parameter:

- Die Übergabe ist hier prinzipiell identisch zur einfachen Suche, jedoch sind in diesem Fall wesentlich mehr Parameter beteiligt.

- **Screenshot:**

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/

angemeldeter Nutzer: Bedlewski, Jana

Mittwoch, der 24. November 2010, 14:11 Uhr

Start * Neues Angebot/Gesuch Meine Angebote/Gesuche Meine Buchungen Kontaktinfos

☒ 6 Angebot(e) gefunden

Frequenz	Datum	Startort	Zielort	von
montags	Mo, 20. Sep 2010	Flöha	Döbeln	J. Bedlewski
montags	Mo, 20. Sep 2010	Flöha	Dresden	S. Weiß
montags	Mo, 20. Sep 2010	Dresden	Flöha	S. Weiß R
montags	Mo, 20. Sep 2010	Leipzig	Flöha	J. Bedlewski
täglich	Di, 22. Jun 2010	Mittweida	Flöha	J. Bedlewski
täglich	Mi, 14. Jan 2009	Mittweida	Dresden	F. Tischer

☒ 1 Gesuch(e) gefunden

Frequenz	Datum	Startort	Zielort	von
einmalig	Fr, 10. Dez 2010	Flöha	Unterföhring	J. Bedlewski

Zurück

Details des Angebotes/Gesuches betrachten
*R - Rückfahrt (auch einzeln buchbar)

Abbildung 4-8 Screenshot Suchergebnisse

- **Funktionsweise:**

(1) Selbstaufruf

Um die Liste der Ergebnisse über einen Selbstaufruf hinaus aufrecht zu erhalten, müssen die Suchparameter in einer Session gespeichert werden. Das ist nötig, da ein Selbstaufruf (Sortierung, Aktivierung) eine erneute Datenbankabfrage auslöst. Hier als Beispiel die Handhabung des Parameters „Art“:

```
...
if ($_POST[R1])
{
    $auswahlart = $_POST[R1];
    $_SESSION[auswahlart]=$auswahlart;
}
else
{
    if ($_SESSION[auswahlart]) $auswahlart =
    $_SESSION[auswahlart];
}
...
```

(2) Aufruf durch Weiterleitung von der Startseite (einfache Suche)

Der Suchalgorithmus ist in mehrere Abschnitte unterteilt, um den strukturellen Überblick zu bewahren. Die einzelnen Teile werden (je nach Bedarf) kontinuierlich wie einzelne Textbausteine aneinander angehängen, was letztendlich die Datenbankabfrage ergibt. Bei dieser einfachen Suchanfrage sind jedoch nur zwei dynamische Parameter beteiligt, wodurch sich das grundlegende Schema hier besser erkennen lässt.

Der allgemeine Teil stellt die Mindestanforderungen an eine solche Abfrage bereit. Das heißt, selbst wenn keine zusätzlichen Angaben getroffen werden, lässt sich so ein gültiger Aufruf realisieren, der alle derzeit aktuellen Eintragungen als Ergebnis liefert. Dieser Teil beinhaltet folgende wichtige Details:

- Die Tabelle der Datenbank, auf die zugegriffen werden soll
- Kriterien zur Filterung abgelaufener Einträge
- Prüfung des Status, um keine blockierten Einträge aufzulisten
- Die Trennung nach Angeboten bzw. Gesuchen
- Den Sortierungsabschnitt

Die Suchparameter selbst betreffend erfolgt bei den Orten ein einfacher Vergleich der getätigten Angaben mit Start- und Zielort, sowie den Zwischenhaltpunkten (nur relevant bei Angeboten).

Bei der Handhabung des Datums reicht eine einfache Überprüfung auf Übereinstimmung nicht aus, da in diesem Fall ein Teil der Eintragungen vernachlässigt werden würde. Der Mitfahrdienst bietet die Möglichkeit auch aus regelmäßigen Angeboten Einzelfahrten zu buchen. Deshalb müssen bei der Suche nach Fahrten eines bestimmten Datums auch alle regelmäßigen Angebote aufgelistet werden, die für den entsprechenden Wochentag gültig sind. Deren eingetragener Zeitpunkt stellt jedoch nicht das Fahrtdatum an sich, sondern den Beginn der Gültigkeit dar, weswegen eine reine Überprüfung auf den Zeitpunkt diese Inserate übersehen würde. Deswegen wird für das bei der Suche eingegebene Datum geprüft, um welchen Wochentag es sich handelt, woraufhin mittels *switch/case*-Anweisung der Datenbankaufruf entsprechend angepasst wird. Die ursprüngliche Prüfung des Datums bleibt Teil der Abfrage, wird aber erweitert um den Wochentag mit dem Kriterium, alle passenden regelmäßigen Angebote bis zum eingegebenen Zeitpunkt zu berücksichtigen. Aber auch bei regelmäßigen Gesuchen ist diese Vorgehensweise durchaus als sinnvoll zu bewerten. Eintragungen dieses Typs müssen immer komplett gebucht werden und sind deshalb nur über die Suche zugänglich, wenn bisher kein Interesse eines anderen Nutzers bestand. Durch diese zusätzliche Aufmerksamkeit erhöhen sich deshalb die Vermittlungschancen.

(3) Aufruf durch Weiterleitung vom Suchformular (Detailsuche)

Durch die Vielzahl an beteiligten Parametern wirkt der Suchalgorithmus zwar etwas komplexer, aber die grundlegende Struktur ist auch hier wieder zu finden. Wegen der Möglichkeit der personenbezogenen Suche muss jedoch zusätzlich die Tabelle des persönlichen Profils mit einbezogen werden.

Bei der Angabe der Identität eines Nutzers werden ein einfacher Vor- und/oder Nachname, sowie eine bestimmte Seminargruppe akzeptiert. Bei der Eingabe eines kompletten Namens muss dieser vollständig mit einem anderen Nutzer übereinstimmen. Wird jedoch beispielsweise nur nach einer Teilbezeichnung gesucht, werden alle Personen berücksichtigt, welche diesen Teil irgendwo in ihrem Namen enthalten. Rauchverhalten und Geschlecht von Fahrer bzw. Mitfahrer werden mittels eines simplen Abgleichs mit den

hinterlegten Daten per Fallunterscheidung ermittelt (*switch/case*-Anweisung). Diese persönlichen Parameter richten sich dabei ausschließlich nach den (teils optionalen) Eintragungen im Profil. Weiterhin reicht der einfache Vergleich auch bei den Parametern Frequenz, sowie bei Start- und Zielort (hierbei selbes Prinzip wie bei der einfachen Suche).

Eine weitergehende Verarbeitung ist jedoch bei Datum und Uhrzeit nötig, aufgrund der möglichen Angabe eines Toleranzbereiches. Dabei müssen zunächst die beiden Parameter für den oberen und unteren Grenzbereich ermittelt werden.

Durch die Verwendung eines Timestamps bei der Handhabung des Datums muss für jeden Toleranzwert nur der äquivalente Zeitwert addiert bzw. abgezogen werden. Dabei hat ein Tag den Wert 86400 (maximal fünf Tage → Wert = 432000). Der entsprechende Teil des SQL-Befehls fragt dann den Bereich zwischen den beiden ermittelten Werten ab. Im zweiten Teil des Abrufs wird (analog zur einfachen Suche) sichergestellt, dass auch die passenden regelmäßigen Inserate mit aufgelistet werden, die schon vor dem gesuchten Datum gestartet sind. Dabei muss hier noch zusätzlich der gewählte Toleranzwert einbezogen werden. Ab einem Wert von 3 sind aber in jedem Fall alle Wochentage beteiligt, weshalb der Bereich von 3 bis 5 nicht mehr in die Fallunterscheidung mit einbezogen werden muss.

...

```
if ($montags == "1"){
    $frequenz_wochentag.="(auswahl_frequenz='montags') OR ";
    $frequenz_werktags = "an"; }

...

if ($frequenz_werktags == "an"){
    $frequenz_wochentag.="(auswahl_frequenz='werktags') OR "; }

...

$db_suche_an.="((ts_dat BETWEEN $datetime_minus AND
    $datetime_plus) OR ...

... ((ts_dat <= $datetime_plus) AND ($frequenz_wochentag
    (auswahl_frequenz='täglich')))) AND ";

...
```

Bei der Variable `$frequenz_wochentag` handelt es sich um den zusammengesetzten String, welcher in angepasster Form die in der Suche abgedeckten Wochentage enthält. Die Frequenz täglich wird immer eingefügt, da diese auf jeden Wochentag zutrifft. Werktags wird erst berücksichtigt, wenn die Suche mindestens einen der Tage von Montag bis Freitag einschließt. Dies trifft auch zu, wenn ein Tag des Wochenendes beteiligt ist, da sich ein Nutzer möglicherweise informieren will, was allgemein für Fahrten an den gewünschten Tagen angeboten werden (es müssen also nicht alle Fahrten für den gesamten Zeitraum gültig sein).

Die Toleranzberechnung der Uhrzeit erfolgt in mehreren Schritten. Zuerst wird der Stundenanteil des eingegebenen Wertes abgeschnitten um danach den entsprechenden Toleranzwert addieren bzw. subtrahieren zu können:

```
...  
  
$zeit_cut = substr($zt,0,2);  
$zeit_zw_plus = $zeit_cut + $auswahluhrzeittol;  
  
...
```

Bei Uhrzeiten muss allerdings zusätzlich darauf geachtet werden, dass ein Tag nur 24 Stunden hat. Deshalb muss im Falle einer Über- bzw. Unterschreitung durch den Toleranzwert das Ergebnis entsprechend korrigiert werden.

```
...  
  
$zeit_zw_plus = $zeit_zw_plus - 24;  
$zeit_zw_minus = 24 + $zeit_zw_minus;  
  
...
```

Wird dabei ein Stundenwert kleiner 10 ermittelt, muss noch eine führende 0 ergänzt werden, um eine korrekte Suche zu ermöglichen. So ergeben sich die Parameter `$plus_full` und `$minus_full`.

Bei der Datenbankabfrage kann in dieser Situation der Befehl `BETWEEN` leider nicht genutzt werden, da es sich bei der Uhrzeiteintragung um keinen einfachen Zahlenwert wie etwa einen Timestamp handelt. Durch die 24 Stunden Grenze muss die Abfrage aus mehreren Teilen zusammengesetzt werden:

```
...  
  
if ($plus_full >= $minus_full)  
{  
    $db_suche_an.="(zeit >= $minus_full AND zeit <=  
    $plus_full) AND ";  
    $db_suche_ge.="(zeit >= $minus_full AND zeit <=  
    $plus_full) AND ";  
}  
else  
{  
    $db_suche_an.="((zeit < 24 AND zeit >= $minus_full) OR  
    ";  
    $db_suche_ge.="((zeit < 24 AND zeit >= $minus_full) OR  
    ";  
  
    $db_suche_an.="(zeit >= 0 AND zeit <= $plus_full)) AND  
    ";  
    $db_suche_ge.="(zeit >= 0 AND zeit <= $plus_full)) AND  
    ";  
}  
  
...
```

Ein kurzes Rechenbeispiel:

- eingegebene Uhrzeit = 22 Uhr / Toleranzwert = 5
- Untergrenze = 17 Uhr / Obergrenze = 3 Uhr
- "zeit BETWEEN 17 AND 3" → unsinnig, da diese Abfrage die 24 Stunden Grenze nicht beachtet
- $(17 < \text{zeit} < 24)$ oder $(0 < \text{zeit} < 3)$ → somit erfolgt eine Trennung des Intervalls an der 24 Stunden Grenze und sobald sich ein gerade überprüfter Zeitpunkt in einem der beiden Teilintervalle befindet, ergibt sich ein Treffer.

Das heißt, mitten am Tag (wenn Obergrenze > Untergrenze) wäre der einfache `BETWEEN` Befehl durchaus ausreichend (siehe *if*-Anweisung). Kommt es jedoch zu einer Über- bzw. Unterschreitung von 24 bzw. 0 Uhr, dann versagt diese Abfrage.

Da bei dieser Suchvariante auch die Art des Inserats unterschieden werden kann, wird letztendlich die Abfrage (und damit die Bearbeitung des abschließenden allgemeinen Teils) nur für den gewünschten Typ durchgeführt (Angebot oder Gesuch oder beides).

4.12 Seite zur Buchung eines Angebotes / Gesuches

- **Dateiname:**

- `formular.MfzBuchung.asp / MfzBuchung.asp / platzcheck.asp`

- **Zweck der Seite:**

Über dieses Formular kann eine bestimmte Fahrt gebucht werden, das heißt es besteht die Möglichkeit aus den angebotenen bzw. gesuchten Sitzplätzen eine bestimmte Anzahl zu reservieren bzw. die gewünschte Anzahl bereitzustellen. Bei Angeboten existieren diverse Zusatzoptionen, wie etwa die (mit leichten Einschränkungen) freie Wahl von Start- und Zielort, oder die je nach Frequenz individuelle Wahl von Fahrzeitpunkt oder –Zeitraum. Bereits vorgenommene Buchungen können über dieses Formular auch aktualisiert werden.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- Wie bei allen Formularen werden auch hier bei jedem *post*- Vorgang einige Parameter *hidden* übergeben, teilweise für die Eingabeprüfung. Hauptsächlich handelt es sich hierbei um Detailangaben des zugehörigen Inserats, welche für die weitergehende Verarbeitung im Hinblick auf das Abspeichern bzw. das Update benötigt werden. Im Buchungsablauf gibt es aber auch Stellen, an denen durch bestimmte Interaktionen ein Selbstaufruf ausgelöst wird, und entsprechende Variablen aktiviert und übergeben werden (diese Variablen sind also nicht bei jedem Selbstaufruf von Bedeutung).

- `$R1` → Auf die Gruppe der Radio-Buttons für die Auswahl der grundlegenden Frequenz (regelmäßig, Auswahl, einmalig) kann über die Variable `$R1` zugegriffen werden (wird nur bei direkter Auswahl übergeben, nicht bei jedem Selbstaufruf).
 - `$platzcheck` → Dabei handelt es sich um die Bezeichnung des Buttons. Wird dieser ausgelöst, erfolgt ein *submit*, worauf eine entsprechende Überprüfung auf die verfügbare Platzanzahl stattfindet (unter Einbeziehung des Quelltextes aus *platzcheck.asp*). Der Button kommt nur bei der Auswahl einzelner Wochentage und der Datumsangabe bei Einzelfahrten zur Anwendung.
 - `$buchen_start` → Bei der Strecke passt sich die Liste des Zielortes der Auswahl des Startortes an, weswegen dieser Parameter bei jedem erneuten Aufruf der Seite inhaltlich geprüft und mit den Inseratsdetails verglichen wird. Die andere Liste passt sich dementsprechend an. Damit soll die identische Wahl beider Orte verhindert werden.

(2) Aufruf aus der Detailansicht eines Angebotes oder Gesuches

→ Parameter:

- `$mitf_iid` → Die ID des zugehörigen Inserats wird für diverse Datenbankabfragen gebraucht, hauptsächlich um die für die Buchung benötigten Details abzurufen. Aber auch beim späteren Abspeichern ist diese ID von Bedeutung, beispielsweise als Referenz, beim Update des Inserats oder dem Absenden der Hinweismail.
- `$back = ins` → Beim Abbruch der Eingabe erfolgt die Weiterleitung zurück zur Detailansicht des jeweiligen Inserats (in Verbindung mit `$mitf_iid`)

(3) Aufruf aus der Buchungsübersicht (nur Angebote → Buchungen für Gesuche können nicht abgeändert werden)

→ Parameter:

- `$mitf_bid` → Die Buchungs- ID wird (analog zu den Inseraten selbst) übergeben, um den entsprechenden Eintrag zu laden.
- `$back = buch_an` → Beim Abbruch der Eingabe wird der Nutzer in diesem Fall zurück zur Übersicht weitergeleitet.
- `$updateopt = an` → Die Variable zeigt an, dass es sich hierbei um ein mögliches Update (statt einem neuen Eintrag) handelt.

▪ **Screenshot:**

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/

angemeldeter Nutzer: Bedlewski, Jana

Mittwoch, der 24. November 2010, 14:17 Uhr

! Sollten keine oder zu wenig Plätze frei sein, dann nutzen sie doch die [Kontaktmöglichkeit](#) zum Inserenten des Angebotes. Möglicherweise können ja noch ein oder zwei zusätzliche Plätze zur Verfügung gestellt werden (als die im Angebot eingetragen) !

Start Meine Angebote/Gesuche Meine Buchungen Meine Bewertungen Kontaktinfos

Buchung vornehmen

Eckdaten der Fahrt

Art: Angebot
Frequenz: täglich
Datum: Mi, 1. Dez 2010
Startort: Oberförhring
Zielort: Unterförhring
Treffpunkt: Pharaohaus
Plätze: 2
Preis pro Platz: Verhandlungssache - Vorschlag: 1 €
Bemerkungen: (keine Angabe)

Plätze buchen

Frequenz: ☐ regelmäßig ☒ Auswahl ☐ einmalig

Wochentage: ☒ mo ☐ di ☐ mi ☐ don ☒ fr ☐ sa ☐ so

Strecke: Startort: Zielort:

Platzanzahl: Plätze: (noch verfügbar: 2)

Plätze buchen:
 Datumsauswahl per Kalender
 rot - Pflichtangabe
 blau - freiwillige Angabe

Abbildung 4-9 Screenshot Buchungsformular

▪ **Funktionsweise:**

(1) Selbstaufruf

Bei diesem Formular gibt es grundsätzlich zwei Situationen, in denen ein Selbstaufruf ausgelöst werden kann. Einerseits handelt es sich dabei wie bei jedem Formular um die Eingabeprüfung, andererseits existieren im Buchungsablauf der Angebote (je nach Frequenz) mehrere Punkte, an denen der Nutzer eine Auswahl treffen muss (worauf sich das Formular entsprechend anpasst).

Die Eingabeprüfung ist im Grunde nur bei Angeboten relevant, da Gesuche nur komplett gebucht werden können, und deshalb ist eine Prüfung in dem Fall nicht vorgesehen ist. Bei einwandfreier Eingabe wird der Nutzer nach dem Abspeichern zur Übersicht weitergeleitet. Durch die diversen Auswahlmöglichkeiten bei den Angeboten gibt es jedoch auch bei diesem Formular Raum für Fehler. Folgende Kriterien werden untersucht:

- Datum muss angegeben werden (Einzelbuchungen)
- Datum muss in der Zukunft liegen
- Datum darf nicht vor dem Startdatum des Inserats liegen
- Datum muss zur gewählten Frequenz passen

- Tag(e) auswählen über die Checkboxes (Auswahl)

Im Buchungsablauf existieren (bis zu) drei Auswahlsschritte, die ein *submit* des Formulars, und damit (laut Definition) einen Selbstaufruf der Seite auslösen.

- *Frequenz (regelmäßig/Auswahl/einmalig)*

Dieser Parameter ist der Ausgangspunkt der Buchung (soweit die Frequenz des Inserats dies unterstützt). Wird einer der Radio-Buttons (`$R1`) ausgewählt, erfolgt der Selbstaufruf -

`onchange="submit(this.form)"` - und der nächste Abschnitt wird angezeigt.

- *Einzeltagesauswahl (Auswahl) / Datumseingabe (einmalig)*

Die einzelnen (regelmäßig zu buchenden) Tage werden über Checkboxes ausgewählt und das Datum wird wie gewohnt in einem Textfeld eingetragen. Beide Optionen müssen aber jeweils noch durch den Button „prüfen“ bestätigt werden, um das *submit* (und damit die Platzberechnung) auszulösen.

- *Strecke und Platzanzahl*

Sind Zwischenhaltepunkte im Angebot eingetragen, gibt es bei der Strecke verschiedene mögliche Kombinationen. Start- und Ziel dürfen aber nicht identisch sein, weswegen sich die Liste der Zielorte entsprechend der Wahl des Startortes anpasst. Hier als Beispiel der Fall, dass als Startort der erste Zwischenhalt gewählt wurde:

...

```
if ($buchen_start == $ueber1)
{
    $ueber11 = "";
    $ueber21 = $ueber2;
    $ueber31 = $ueber3;
}

...

$obj->props[Buchung][buchen_ziel]-
>SelectInhaltZiel($obj, $ziel_inserat, $ueber11,
$ueber21, $ueber31)
```

...

Bei jeder Auswahl eines Startortes erfolgt ein *submit*

(`onchange="submit(this.form)"`) und so wird die Liste der Zielorte passend neu geladen.

(2) Aufruf aus der Detailansicht eines Angebotes oder Gesuches

In diesem Fall wird das Formular für einen neuen Eintrag bzw. die erstmalige Buchung dieses Inserats geladen. Mit der ID des Inserats (`$mitf_iid`) werden die für die Buchung benötigten Details abgerufen, wie etwa Start- und

Zielort, die Zwischenhaltepunkte und die Anzahl der freien Sitzplätze pro Wochentag (für die Berechnung der letztendlich verfügbaren Plätze benötigt).

Wichtig für die spätere Eintragung in die Datenbank ist auch die Prüfung auf eine mögliche Rück- oder Hinfahrt und ob diese schon gebucht wurde. Denn beide Fahrten werden als Einheit im Sinne der Bewertung betrachtet, und deshalb wird zwar in beiden Fällen ein temporärer Bewertungseintrag erstellt, aber nur einer von beiden wird freigegeben (und zwar derjenige, dessen zugehörige Fahrt zuerst stattfindet - damit nach der ersten Fahrt schon bewertet werden kann). Dabei wird der Parameter `$bew_block` entweder auf 0 (Freigabe) oder 1 (gesperrt) gesetzt. Bei regelmäßigen Inseraten ist es außerdem relevant zu prüfen, ob der aktuelle Eintrag von demselben Nutzer vorher schon einmal gebucht wurde, und noch ein gültiger Bewertungsdatensatz existiert. Da es sich in diesem Fall um dieselbe Fahrt handelt, soll die spätere Bewertung nur als Update der Bisherigen geführt werden. Genauso wird der wegen der offenen Bewertung noch bestehende zugehörige Buchungseintrag selbst nur passend aktualisiert, um unnötige Datensätze zu vermeiden.

Je nach Frequenz des Inserats hat der Nutzer verschiedene Optionen zur Verfügung:

- *Gesuche*
 - im Ist-Zustand zu buchen (keine Auswahlmöglichkeit)
- *einmalig*
 - Strecke + Plätze (Selbstaufruf bei Ortswahl)
- *täglich/werktags*
 - Frequenz (regelmäßig/Auswahl/einmalig)
 - Auswahl Einzeltage/Datum (durch „prüfen“ bestätigen)
 - Strecke + Plätze
- *Einzeltage (montags, dienstags etc.)*
 - Frequenz (regelmäßig/einmalig)
 - Auswahl Datum (durch „prüfen“ bestätigen)
 - Strecke + Plätze

Nach dem *post*- Vorgang des Formulars wird zunächst auf Fehler in der Eingabe geprüft. Ohne Beanstandung erfolgen die objektbasierte Speicherung und danach die weitergehende Abarbeitung des Quelltexts, wobei unterschieden wird, ob es sich um ein Update handelt oder nicht. Aber im Grunde wird zunächst einmal (frequenzabhängig) die aktualisierte Anzahl der freien Plätze für die einzelnen Wochentage berechnet, um anschließend ein Update des Inseratseintrages durchzuführen.

...

```
$mon_det = $row_details["montag"];
```

...

```
$mon_up = $mon_det - $_POST[buchen];
```

...

(dieser Codeabschnitt betrifft eine erstmalige Buchung - das Prinzip ist bei Updates jedoch identisch)

Danach wird eine Benachrichtigungsmail an den Inserenten geschickt, und der temporäre Bewertungseintrag erstellt.

Wichtig für die spätere Handhabung beim Löschen des zugehörigen Inseratseintrags ist außerdem die Ermittlung des ersten möglichen Gültigkeitstags der Fahrt. Dabei wird die Variable `$fahrt_tag1` gebildet, oder direkt übernommen wenn es sich um eine Einzelfahrt oder ein Gesuch handelt. Dieser Parameter ist ausschlaggebend darüber, ob der Eintrag direkt gelöscht werden kann, oder wegen späterer Verwendung nur gesperrt werden muss.

(3) Aufruf aus der Buchungsübersicht

Durch die Übergabe der ID des jeweiligen Buchungsvorgangs wird das Formular mit den entsprechenden Daten geladen, und der folgende Speichervorgang resultiert in einem Update. Bei der Handhabung der Daten gibt es leichte Änderungen, beispielsweise muss die Anzahl der noch verfügbaren Plätze um die bisher gebuchte Anzahl ergänzt werden (anstatt der unmittelbaren Verwendung der Inseratsdaten).

...

```
$mon_frei = $mon_det + $mon_old;
```

...

Des Weiteren wird nach der abgeschlossenen Modifikation der temporäre Bewertungseintrag aktualisiert, und auch die Mitteilung per E-Mail weißt auf eine Änderung hin.

(A) *platzcheck.asp*

Diese Datei ist in *MfzBuchung.asp* implementiert, und wird somit beim Aufruf des Formulars mit geladen. Es handelt sich dabei eigentlich nur um die Auslagerung eines Teils des Quelltextes, welcher für die Berechnung der freien Plätze zuständig ist. Das heißt, immer wenn diese Anzahl aufgrund einer Interaktion mit dem Formular ermittelt werden muss - Wahl von Frequenz, Einzeltagen oder Datum – kommt dieser Abschnitt zum Einsatz. Wegen des gro-

ßen Umfangs wurde diese zusammenhängende Einheit in eine separate Datei ausgelagert, um eine bessere Übersichtlichkeit im Formular selbst zu bewahren. Der Aufbau erscheint auf den ersten Blick möglicherweise etwas unübersichtlich, jedoch steckt ein strukturiertes System dahinter, was in folgenden Ebenen angeordnet ist:

- *Frequenz Inserat* (außer Gesuche/einmalige Einträge → da festgelegte Anzahl)
 - *Einmalige Buchung vorhanden* (`$count_buch_check_einmalig > 0`)
 - Frequenz Formular – *einmalig* (`$_POST[R1]=="einmalig"`)
 - Frequenz Formular – *Auswahl* (`$_POST[R1]=="auswahl"`)
 - Frequenz Formular – *regelmäßig* (`$_POST[R1]=="regelm"`)
 - *Keine einmalige Buchung vorhanden* (else)
 - Frequenz Formular – *einmalig* (`$_POST[R1]=="einmalig"`)
 - Frequenz Formular – *Auswahl* (`$_POST[R1]=="auswahl"`)
 - Frequenz Formular – *regelmäßig* (`$_POST[R1]=="regelm"`)

Einfach nur die Zahl der freien Plätze auszulesen wäre bei regelmäßigen Inseraten nicht akkurat genug. Liegen etwa Einzelbuchungen vor, dürfen die dabei reservierten Plätze nicht für die Gesamtdauer des Angebotes als belegt gelten. Deswegen werden bei allen Buchungsvorgängen mit der Wahl einer regelmäßigen Frequenz die einmalig reservierten Plätze wieder dazugezählt (ein entsprechender Hinweis wird dem Buchenden eingeblendet). Dies gilt auch für Einzelfahrten, welche am selben Wochentag, jedoch an einem anderen Datum stattfinden sollen (währen also für das gewünschte Datum nicht relevant und können somit wieder einbezogen werden). Um dies zu realisieren, wird mittels einer Datenbankabfrage nach bestehenden Einzelbuchungen gesucht, und pro Durchgang der *while*- Schleife...

...

```
while
(
    $row_buch_check=mysql_fetch_array($result_buch_check)
)
{
```

...

...werden die passenden Details ausgelesen und bei Übereinstimmung entsprechend verarbeitet.

Als Beispiel folgende Situation: Der Nutzer möchte an einem bestimmten Montag von Chemnitz nach Altenburg fahren. Das einzige Inserat für diese Strecke ist ein als regelmäßig werktags eingetragenes Angebot mit 5 zu vergebenden Sitzplätzen. Weil ab und zu noch weitere Personen in diesen Ort zum dortigen Flughafen fahren möchten, liegen bereits zwei Einzelbuchungen vor, wovon eine auf denselben Tag fällt. Zusätzlich ist auch noch eine komplette Buchung eingetragen. Demnach ergibt sich folgende Rechnung:

- Die Variable `$frei` wird hier beim ersten Durchgang der *while*-Schleife initiiert und mit einem Zwischenwert belegt. Der gerade behandelte Datensatz ist in diesem Fall die Einzelfahrt, welche am selben Tag stattfindet → `$instime = $dattime` (Datum Buchungseintrag = Datum momentaner Buchungsvorgang). Deshalb besitzt der im Inserat eingetragene Ausgangswert Gültigkeit (hier kein Ausgleich durch dazuaddieren nötig):

...

```
if (!$frei){ $frei = $mon_frei; }
```

...

`$mon_frei` beinhaltet in diesem Fall den Wert 2 (5 als Startwert - 3 insgesamt reservierte Plätze), wodurch bei der erstmaligen Belegung auch `$frei` diesen Wert annimmt.

- Beim zweiten Schleifendurchgang muss die bei dem Buchungseintrag für die zweite Einzelfahrt reservierte Zahl an Sitzplätzen der bisher ermittelten Variable `$frei` hinzugefügt werden, da ein anderes Fahrtdatum zu Grunde liegt:

...

```
$buch_zw = $row_buch_check["buchen"];
```

...

```
$frei = $frei + $buch_zw;
```

...

Damit erhöht sich der Wert der Variable `$frei` auf 3.

Diese Variable kann im Allgemeinen durchaus mehrere Male die Schleife durchlaufen. Falls bei einem weiteren Durchlauf anschließend die in Schritt 1 beschriebene Situation (erneut) eintritt, darf der hier gebildete Wert nicht einfach wieder mit dem Ausgangswert des Inserats überschrieben werden. Um die bereits verarbeitete Variable in dieser Form zu erhalten, werden die beiden Werte verglichen, und der größere von Beiden (was der Addierte sein muss) wird beibehalten:

...

```
if ($frei > $mon_frei){$frei = $frei;}else{$frei = $mon_frei;}
```

...

-
- Der durch die regelmäßige Buchung belegte Platz ist hier (ähnlich zu Schritt 1) in der Variable `$mon_frei` bereits berücksichtigt und muss nicht zusätzlich verarbeitet werden. Deshalb ergibt sich eine endgültige Anzahl von 3 freien Plätzen für den angegebenen Montag (`$frei` wird nach der Berechnung auf `$plaetze_frei` bzw. `$plaetze_offen` abgebildet).

Bei der Frequenzoption „Auswahl“ finden sich diese eben erläuterten Mechanismen im ersten Berechnungsschritt in leicht angepasster Form wieder. Jedoch muss im zweiten Schritt noch aus allen relevanten Werten (d.h. gewählten Wochentagen) die Anzahl der durchgängig garantiert verfügbaren Sitzplätze ermittelt werden. Dafür wird aus den entsprechenden Variablen der kleinste Wert (= Mindestwert) ermittelt:

```
...  
    $plf = $plaetze_mon;  
    ...  
    if ($plf < $plaetze_die){$plf = $plf;}else{$plf =  
    $plaetze_die;}  
    ...
```

Diese Vorgehensweise wird auch bei der Option „regelmäßig“ angewendet, wenn die Frequenzen mehrere Tage beinhalten (betrifft täglich und werktags).

Bei jeglichen Buchungsvorgängen wird die Anzahl der noch verfügbaren Plätze im Inseratseintrag reduziert, weshalb es wegen der Einzelbuchungen durchaus auch zu einer negativen Menge im Datensatz kommen kann. Bei der Berechnung der freien Plätze wird diese Diskrepanz durch das Rückaddieren wieder ausgeglichen.

Die Variable `$plaetze_frei` (bzw. `$plaetze_offen` bei Updates) wird letztendlich bei diesen Durchläufen gebildet und an die zuständige Funktion zur Ausgabe in einer Auswahlliste weitergegeben:

```
...  
    $obj->props[Buchung][buchen]->SelectInhaltPlatz($obj,  
    $plaetze_frei)  
    ...
```

Die hier vorliegende Umsetzung der Buchungsfunktion war ursprünglich nicht so geplant; es wurde erst eine recht simple Variante integriert, welche an die Realisierung in der vorhergehenden Version des Mitfahrdienstes angelehnt war. Jedoch hat es sich letztendlich nach spontanen Überlegungen als konsequenter und passender zu den vorher gewonnenen Erkenntnissen erwiesen, diese Lösung mit erweitertem Funktionsumfang zu entwickeln. Bei der eher kurzfristigen Realisierung kam es ursprünglich darauf an, ohne übermäßigen Zeitaufwand ein funktionsfähiges Resultat zu programmieren (wobei die übergeordnete Planung etwas in den Hintergrund trat). Dabei wurden die einzelnen Segmente ausgehend von einem groben Grundgerüst laufend ergänzt und erweitert, was den weitläufigen Aufbau erklärt. Ein Großteil der Abschnitte wurde dabei im Grunde für jede Frequenz angepasst und neu eingefügt.

Dadurch hat sich zwar der Umfang ausgedehnt, aber in dem Zusammenhang konnte die Funktionalität so besser sichergestellt werden, als bei dem Versuch alles noch zu rationalisieren (nach dem Motto: funktioniert ein Abschnitt, sollte der Rest nach diesem Vorbild auch laufen...).

4.13 Seiten für die Übersicht vorgenommener Buchungen

- **Dateiname:**

- *MfzBuchungUebersicht.asp*
- *MfzBuchungUebersichtAngebot.asp* / *MfzBuchungAngebot.asp*
- *MfzBuchungUebersichtGesuch.asp* / *MfzBuchungGesuch.asp*
- *MfzBuchungUebersichtAngebotOther.asp* /
MfzBuchungAngebotOther.asp
- *MfzBuchungUebersichtGesuchOther.asp* /
MfzBuchungGesuchOther.asp
- *MfzBuchungUebersichtAngebotIns.asp* / *MfzBuchungAngebotIns.asp*
- *MfzBuchungUebersichtGesuchIns.asp* / *MfzBuchungGesuchIns.asp*

- **Zweck der Seite:**

Ausgehend von der Hauptübersichtsseite *MfzBuchungUebersicht.asp* hat der Nutzer alle Buchungen im Blick - sowohl die selbst vorgenommenen, als auch die für ein selbst erstelltes Inserat vorliegenden. Eigene Buchungen können geändert, gelöscht und im Detail betrachtet werden. Bei Fremdbuchungen ist es nur möglich, die zugehörigen Informationen abzurufen. Wegen der besseren Übersichtlichkeit, wurden die einzelnen Inhalte auf separate Seiten ausgelagert, wobei *MfzBuchungUebersicht.asp* als Ausgangspunkt genutzt wird. Bei einem selbst erstellten Angebot oder Gesuch kann über einen Link auf der Detailseite des Inserats auch direkt auf die entsprechende(n) Buchung(en) zugegriffen werden.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- Der Selbstaufruf betrifft nur die jeweiligen Unterseiten, auf denen letztendlich die Tabellen zu finden sind. Diese kann der Nutzer sortieren oder unter Zuhilfenahme von `DbPage()` auch an- bzw. ausschalten, wobei zur Kennzeichnung der durchzuführenden Aktionen bestimmte Parameter übergeben werden.

- Der jeweilige `$REMOTE_USER` sieht immer nur seine eigene Buchungsübersicht, deshalb ist es hier im Gegensatz zur Übersicht der Angebote und Gesuche nicht nötig eine Nutzer ID separat zu übergeben. Auch die zur Sortierung verwendete Session funktioniert bei dieser Anwendung ohne übergebene Identifizierung einwandfrei.

- `$sortan`, `$orderan`, `$richtungen` → Bei der Sortierung der Tabelle gebuchter Angebote wird die Variable `$sortan` durch Anklicken des Titels der entsprechenden Spalte im Selbstaufruf übergeben. `$orderan` steuert bei der Datenbankabfrage die Sortierung, indem der übergebene Parameter mit der Datenbanktabelle verknüpft wird, auf welche die Sortierung angewendet werden soll. Über `$richtungen` wird die jeweilige Sor-

tierungsrichtung angezeigt, welcher immer abwechselnd ab- und aufsteigend ist.

- \$sortge, \$orderge, \$richtungge → Sortierung der durch den Nutzer gebuchten Gesuche
- \$sortano, \$orderano, \$richtungano → Sortierung der Buchungen eigener Angebote
- \$sortgeo, \$ordergeo, \$richtunggeo → Sortierung der Buchungen eigener Gesuche
- \$zeigeame, \$zeigeame, \$zeigeanother, \$zeigeanother → Zum Ein- und Ausblenden der Tabellen wird diese Variable als Statusmeldung der verantwortlichen Checkbox übergeben.

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

→ Parameter:

Da diese Ansicht nur für den \$REMOTE_USER ausgelegt ist, wird keine separate Nutzeridentifizierung verlangt. Im Gegensatz zu einem Formular existiert hier auch keine Notwendigkeit für einen Abbruch (und dem damit benötigten Erkennungsparameter), ein einfaches Zurückgehen im Browser oder die Verwendung der angebotenen Linkleiste sind die Möglichkeiten der Navigation (gilt auch für Punkt (3)).

(3) Aufruf über die Detailseite eines Inserats

→ Parameter:

- \$mitf_iid → Aus der Inseratsansicht heraus wird der Schlüssel des entsprechenden Angebotes bzw. Gesuches übergeben, da es sich hierbei um eine auf diesen speziellen Eintrag angepasste Übersicht handelt (anstatt alle Buchungen aller Inserate eines Nutzers aufzulisten).

▪ Screenshot:

Buchungsübersicht Hochschule Mittweida (FH) University of Applied Sciences

Ⓜ-Login: jbedlews/ [Nach oben] [von ihnen gebuchte Angebote] [von ihnen gebuchte Gesuche]
[bei ihnen gebuchte Angebote] [bei ihnen gebuchte Gesuche]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:18 Uhr

◀ Start [Ⓜ] Neues Angebot/Gesuch [Ⓜ] Meine Angebote/Gesuche [Ⓜ] Meine Bewertungen [Ⓜ] Kontaktinfos

Buchungen:

von Ihnen gebucht	<ul style="list-style-type: none">1 gebuchte(s) Angebot(e)0 gebuchte(s) Gesuch(e)	Angebote/Gesuche die Sie bei anderen Personen gebucht haben
bei Ihnen gebucht	<ul style="list-style-type: none">2 gebuchte(s) Angebot(e)0 gebuchte(s) Gesuch(e)	eigene Angebote/Gesuche die von anderen Personen gebucht wurden

Zurück

Abbildung 4-10 Screenshot Buchungsübersicht (Hauptseite)

von ihnen gebuchte Angebote

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/ [Nach oben]

angemeldeter Nutzer: Bedlewski, Jana
Mittwoch, der 24. November 2010, 14:18 Uhr

Start
Neues Angebot/Gesuch
Meine Angebote/Gesuche
Meine Buchungen
Meine Bewertungen

1 Angebot(e) - von ihnen gebucht

Frequenz	Datum	Startort	Zielort	von	
montags	Mo, 20. Sep 2010	Flöha	Dresden	S. Weiß	<div> <div></div> <div></div> <div></div> <div></div> </div>

Zurück

Buchung ändern/bearbeiten
Details der Buchung betrachten
Details des zugehörigen Angebotes betrachten
Buchung löschen

Abbildung 4-11 Screenshot Buchungsübersicht (fremde Angebote)

▪ Funktionsweise:

(1) Selbstaufruf

Ein Selbstaufruf kommt zustande, wenn der Nutzer die Tabelle sortiert, oder ein- bzw. ausblendet. Zum Aufrechterhalten der kritischen Parameter der Sortierung über die wiederholten Aufrufe hinweg, werden diese in einer Session gespeichert. Der kritische Sortierungsparameter der selbst gebuchten Angebote beispielsweise ergibt sich wie folgt:

```
...
$_SESSION[orderan]= "mitf_buchung.".$sortparam_an;
```

...

(beim Auslösen des Sortierungsvorganges durch Klick auf den Tabellenkopf wird ein Zahlenwert übergeben, der dann in den konkreten Sortierungsparameter – hier `$sortparam_an` – umgesetzt wird)

Der Richtungsparameter ist alternierend, das heißt bei jedem Aufruf wechselt der Zustand, um die Auf- bzw. Abwärtssortierung zu realisieren.

...

```
if(!$_SESSION[richtungan]) $_SESSION[richtungan]="DESC";
else $_SESSION[richtungan]="";
```

...

In die konkrete Datenbankabfrage werden diese Parameter dann wie folgt eingefügt:

...

```
... ORDER BY $_SESSION[orderan] $_SESSION[richtungan] ";
```

...

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

Ohne bisherige Interaktion wird die beim erstmaligen Aufruf generierte Tabelle ausgangs nach dem gebuchten Datum der Fahrt (erster Tag) - `ts_dat` - in absteigender Reihenfolge sortiert dargestellt. Beim Auslesen der benötigten

Informationen aus der Datenbank sind hierbei zwei verschiedene Tabellen beteiligt (Die Buchungs- und die Inseratstabelle), da nicht alle relevanten Informationen in dem Datensatz der Buchung festgehalten sind.

Für diesen Aufruf sind folgende Dateien relevant:

- MfzBuchungUebersicht.asp → „Startseite“
- MfzBuchungUebersichtAngebot.asp / MfzBuchungAngebot.asp → durch den Nutzer gebuchte Angebote
- MfzBuchungUebersichtGesuch.asp / MfzBuchungGesuch.asp → durch den Nutzer gebuchte Gesuche
- MfzBuchungUebersichtAngebotOther.asp / MfzBuchungAngebotOther.asp → von dem Nutzer gebuchte Angebote
- MfzBuchungUebersichtGesuchOther.asp / MfzBuchungGesuchOther.asp → von dem Nutzer gebuchte Gesuche

(3) Aufruf über die Detailseite eines Inserats

Für diesen Aufruf werden folgende Dateien benötigt:

- MfzBuchungUebersichtAngebotIns.asp / MfzBuchungAngebotIns.asp → Buchungen für ein bestimmtes Angebot des Nutzers
- MfzBuchungUebersichtGesuchIns.asp / MfzBuchungGesuchIns.asp → Buchung für ein bestimmtes Gesuch des Nutzers

In diesem Fall handelt es sich ausschließlich um die Buchungen anderer Personen, deshalb ist eine keine Bearbeitung der Reservierung(en) vorgesehen, sondern nur eine Betrachtung der Details.

4.14 Seite zur Darstellung der Details einer Buchung

- **Dateiname:**
 - *MfzBuchungDetails.asp / MfzBuchung.asp*
- **Zweck der Seite:**

Auf dieser Seite werden alle relevanten Details eines bestimmten Buchungsvorgangs dargestellt.
- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf aus der Buchungsübersicht

→ Parameter:

 - \$mitf_bid → Die ID des zugehörigen Buchungseintrags für den Aufruf der Daten.
- **Screenshot:**



Abbildung 4-12 Screenshot Buchungsdetails

- **Funktionsweise:**

(1) Aufruf aus der Buchungsübersicht

Egal ob die Detaildarstellung eines bestimmten Buchungsvorgangs aus der Hauptübersicht (beinhaltet alle Buchungen) oder aus der inseratsbezogenen Variante angewählt wird, der Aufruf und die damit verbundene Verarbeitung sind jeweils identisch. Start- und Zielort, sowie die Anzahl der gebuchten Plätze werden über die Objektfunktion angezeigt. Die Darstellungen des Datums müssen durch Umwandlung des jeweilig ausgelesenen Timestamps noch angepasst werden. Auch die Anzeige der Frequenz wird zur besseren Anschaulichkeit noch passend aufgearbeitet, indem bei der Option „Auswahl“ die jeweiligen Wochentage hinzugefügt werden und der Vermerk „regelmäßig“ bei allen passenden Frequenzen abgesehen von „Auswahl“ ergänzt wird. Ergänzend zu den buchungsspezifischen Informationen hat man Zugang zu den Profildaten des Buchenden oder des Inserenten (je nachdem welche der Personen die Detailseite betrachtet). Nur die an der Buchung beteiligten Personen haben Zugang zu dieser Seite. Dieser wichtige Punkt wird in der Eingangsprüfung untersucht, ebenso wie die Gültigkeit der übergebenen ID sowie des Profils von \$REMOTE_USER.

4.15 Seite zum Löschen einer Buchung

- **Dateiname:**

MfzBuchungDelete.asp

- **Zweck der Seite:**

Diese Seite fungiert als Zwischenschritt beim Löschen einer bestehenden Buchung. Der Nutzer hat so nochmals die Möglichkeit, sich dagegen zu entscheiden. Dies soll einem versehentlichen Löschvorgang vorbeugen.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf von der Übersichtsseite

→ Parameter:

- \$mitf_bid → Hierbei handelt es sich um den Schlüssel des zu löschenden Eintrags.

- **Screenshot:**



Abbildung 4-13 Screenshot Löschen einer Buchung

- **Funktionsweise:**

(1) Aufruf von der Übersichtsseite

Auch bei diesem Datensatz wird die Gültigkeit des Aufrufs geprüft, d.h. ist der Eintrag noch aktiv, wurde die Buchung vom \$REMOTE_USER getätigt, und ist dessen Profil gültig ?

Ein paar Schlüsseldetails zur korrekten Identifizierung der Fahrt werden nochmals aufgelistet (Frequenz, Datum, Strecke gebuchte Platzanzahl). Dazu werden die benötigten Informationen ausgelesen, und teilweise in bearbeiteter Form dargestellt. Das Datum wird aus dem Timestamp rekonstruiert, und die gebuchte Frequenz wird gegebenenfalls textlich aufbereitet. Dabei werden im Falle der Frequenz „Auswahl“ die gebuchten Tage ergänzt und bei allen anderen Buchungen, die keine Einzelfahrt darstellen, zur besseren Erkennbarkeit der Vermerk „regelmäßig“ hinzugefügt.

Wurde der Löschvorgang vom Nutzer bestätigt, werden zuerst die reservierten Plätze auf den Inseratseintrag zurückgerechnet (die Plätze werden ja wieder freigegeben). Danach erfolgt je nach Zeitpunkt der Aufhebung der Bu-

chung (vor oder nach dem ersten möglichen Fahrttag) entweder die Entfernung oder die Sperrung des Datensatzes. Genau wie bei dem Löschen eines Inserats wird auch hier darauf geachtet, den Buchungseintrag und den zugehörigen temporären Bewertungsdatensatz letztendlich direkt zu entfernen, wenn bereits ein derzeit für beide Nutzer aktiver Bewertungsdatensatz der ebenfalls gebuchten Hin- bzw. Rückfahrt existiert. Wurden ursprünglich Hin- und Rückfahrt gebucht, muss außerdem sichergestellt werden, dass der korrespondierende temporäre Bewertungseintrag freigegeben ist, falls die mit diesem Löschvorgang betroffene Fahrt noch nicht stattgefunden hat (ansonsten erfolgt die Freigabe später im Zuge des Bewertungsvorgangs). Dies ist nötig, da beide Fahrten ja als Einheit nur einmal und nicht getrennt bewertet werden können, weswegen der Datensatz der zweiten gebuchten Fahrt automatisch blockiert wurde. Abschließend wird dem Urheber des zugehörigen Inserats eine Mitteilung über die Aufhebung der Buchung per E-Mail geschickt.

4.16 Seite zur Bewertung eines Fahrers / Mitfahrers

- **Dateiname:**

- *formular.MfzBewertung.asp / MfzBewertung.asp*

- **Zweck der Seite:**

Fahrer und Mitfahrer können sich mittels dieses Formulars gegenseitig bewerten. Zuverlässigkeit und Auftreten werden über eine Punkteskala beurteilt und ein kurzer Kommentar ermöglicht ergänzende Anmerkungen.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- Diverse Parameter werden für die Verarbeitung nach dem Speichern benötigt, und deshalb *hidden* übergeben. Dies geschieht bei jedem POST Vorgang, jedoch sind folgende Parameter relevant, da sie für den Selbstaufruf verantwortlich sind (erfolgt nur bei fehlerhafter Eingabe):

- \$R1 → Diese Variable bezeichnet die Gruppe der Radiobuttons, welche für die Bewertung der Zuverlässigkeit zuständig sind. Da es sich hierbei um eine Pflichteingabe handelt, wird diese Variable unter anderem für die Fehlerprüfung benötigt, welche über einen Selbstaufruf entscheidet.
 - \$R2 → Die Beurteilung des Wertes Auftreten erfolgt über Radiobuttons mit dieser Bezeichnung. Genau wie bei \$R1 ist eine Angabe erforderlich.

- (2) Aufruf aus der Übersicht für freigegebene Bewertungen

- Parameter:

- Für den Abbruch der Eingabe muss in diesem Fall keine Variable übergeben werden, da es nur eine feste Stelle gibt, an die zurückgegangen werden müsste.

- \$mitf_bwid → Hierbei handelt es sich um die ID des temporären Bewertungseintrags. Damit werden die Identifizierungsschlüssel des zugehörigen Inserats- und auch des Buchungseintrags ausgelesen, um das Abrufen der zur Darstellung und späteren Abspeicherung benötigten Zusatzinformationen zu ermöglichen. Außerdem erfolgt bei erneuter Bewertung einer regelmäßigen Fahrt nach dem Abspeichern ein Update des temporären Eintrags, wofür diese ID vorhanden sein muss.

▪ **Screenshot:**

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/

angemeldeter Nutzer: Bedlewski, Jana

Mittwoch, der 24. November 2010, 14:25 Uhr

Start Neues Angebot/Gesuch Meine Angebote/Gesuche Meine Buchungen Meine Bewertungen

Bewertung von Sebastian Weiß als Fahrer(in)

Eckdaten der Fahrt

Art: Gesuch
Frequenz: einmalig
Datum: Mo, 20. Sep 2010
Startort: Flöha
Zielort: München
Treffpunkt: Sparkasse Plau
gebuchte Plätze: 2
Preis pro Platz: Verhandlungssache - Vorschlag: 10 €
Bemerkungen: (keine Angabe)

Fahrt bewerten

Zuverlässigkeit: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
Auftreten: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
Kommentar:

Abbrechen Speichern

Fahrt bewerten:
rot - Pflichtangabe
blau - freiwillige Angabe

Abbildung 4-14 Screenshot Bewertungsformular

▪ **Funktionsweise:**

(1) Selbstaufruf

Dieser Aufruf erfolgt nur bei negativer Eingabeprüfung. Wurde keine Angabe bei den Bewertungsparametern Zuverlässigkeit und/oder Auftreten gemacht, ist die Abspeicherung blockiert und der Nutzer wird nach dem Selbstaufruf nochmals auf diese Pflichtparameter hingewiesen.

(2) Aufruf aus der Übersicht für freigegebene Bewertungen

Die Eingangsprüfung nach dem Aufruf der Seite beinhaltet neben der standardmäßigen Begutachtung auf Gültigkeit der übergebenen ID und des Nutzerprofils außerdem die Prüfung auf Beteiligung des `$REMOTE_USER` an der Buchung und/oder des Inserats (nur wenn der Nutzer gebucht hat oder der Inserat ist besteht Zugang). Weiterhin wird darauf geachtet, dass die Bewertung (vom Zeitpunkt her) momentan überhaupt vorgenommen werden darf.

Die ausgelesene Inserats-ID (hier als `$iid` zugewiesen) wird dazu genutzt, benötigte Informationen abzurufen. Zum einen werden im ersten Teil der Seite Details zur Fahrt wiedergegeben, zum anderen richten sich bestimmte Aspekte der weitergehenden Bearbeitung nach diversen Parametern des Inserats.

- Mitfahrer bewertet Fahrer → Start-/Zielort (im endgültigen Bewertungsdatensatz) wird aus Buchung übernommen
- Fahrer bewertet Mitfahrer → Start-/Zielort (im endgültigen Bewertungsdatensatz) wird aus Inseratseintrag übernommen
- regelmäßige Fahrt ? → bei Update wird nächstmöglicher Zeitpunkt für Bewertung berechnet; der temporäre Datensatz wird nötigenfalls für

weitere Bewertungen gesperrt und der korrespondierende Eintrag von Hin-/Rückfahrt freigegeben; temporärer Bewertungseintrag, Buchungseintrag, Inseratseintrag werden automatisch gelöscht, wenn nicht mehr benötigt.

- einmalige Fahrt ? → nach abgegebener Bewertung werden der temporäre Bewertungseintrag, sowie der Datensatz der Buchung und der Fahrt selbst gelöscht (Löschung bei Fahrt nur, wenn die Gegenseite bereits bewertet hat).

Des Weiteren erfolgt auch die Zuweisung einiger Inseratsdetails auf den Bewertungsdatensatz (Start-/Zielort, Datum, Frequenz, Inserent), damit dort ein paar Informationen zur Fahrt enthalten sind (für die Detailanzeige).

Bei erfolgreich abgegebener Bewertung wird der Inserent entsprechend benachrichtigt. Die dabei angewendete Mail-Funktion greift auch auf wichtige Details des Angebotes bzw. Gesuches zurück.

Das Formular selbst wird immer leer initialisiert, da keine direkten Updates vorgesehen sind. Die Aktualisierung bei regelmäßigen Fahrten läuft darauf hinaus, dass jeweils der Durchschnitt aus dem bisherig abgespeicherten und dem aktuell gewählten Wert gebildet wird.

...

```
$zuv_up = (($zuv + $_POST[R1]) / 2);  
$auf_up = (($auf + $_POST[R2]) / 2);
```

...

Für die mögliche Darstellung des Bewertungsschnitts eines Nutzers wird nach jeder Beurteilung der aktualisierte Wert im persönlichen Profil abgespeichert (bei Unterscheidung der beiden Typen Fahrer bzw. Mitfahrer). Das ist einfacher, als an jeder Stelle, an welcher der Wert benötigt wird, den Durchschnitt immer neu zu bilden. Alle bisherigen Bewertungen des gerade aktiven Nutzers werden in einer *while*-Schleife abgearbeitet, um daraus den Mittelwert zu erstellen.

Ein kurzes Rechenbeispiel mit dem Parameter „Auftreten“ für den Nutzer als Mitfahrer:

- erster Durchlauf → (bisheriger) Gesamtwert: 0 – aktueller Wert: 5 – Zähler: 1

...

```
$auf_mit_ges = 0;  
$auf_mit_akt = $row_bew_final["auftreten"];  
$count_mit = 1;
```

...

- zweiter Durchlauf → Gesamtwert: 5 (0+5) – aktueller Wert: 4 – Zähler: 2

...

```
$auf_mit_ges = $auf_mit_ges + $auf_mit_akt;  
$auf_mit_akt = $row_bew_final["auftreten"];  
$count_mit = $count_mit + 1;
```

...

- dritter Durchlauf (selber Abschnitt) → Gesamtwert: 9 (5+4) – aktueller Wert: 3 – Zähler: 3
- nach der Schleife → Gesamtwert: 12 (9+3) / Zähler: 3 = 4 (→ Mittelwert)

...

```
$auf_mit_ges = (($auf_mit_ges +  
$auf_mit_akt) / $count_mit);
```

...

Sollten Buchungs- und/oder Inseratseintrag durch den Status 1 gesperrt sein (weil bereits gelöscht/abgelaufen) – es würde sich hierbei also um die letztmögliche Bewertung dieses Eintrags handeln – wird der entsprechende *next_bew* Parameter auf 0 gesetzt und bei Bedarf im Gegenzug der derzeit gesperrte Bewertungsdatensatz von Hin- oder Rückfahrt für diesen Nutzer freigegeben. Diese Freigabe ist aber nicht relevant für einmalige Fahrten, da in diesem Fall sowieso nur eine einzige Bewertung möglich ist und deshalb die korrespondierenden Datensätze (Buchung und temporäre Bewertung) sofort gelöscht werden können. Dies gilt auch für den zugehörigen Inseratseintrag selbst, insofern dieser abgelaufen ist und keine weitere Buchung vorliegt.

Steht der Parameter *next_bew* nach der aktuellen Bewertung für beide Nutzer auf 0, werden der zugehörige Buchungs- und der temporäre Bewertungsdatensatz entfernt. Der Inseratseintrag wird auch in diesem Fall direkt gelöscht, wenn dieser keine zusätzlichen Buchungen und keine Gültigkeit mehr besitzt.

Ist keine weitere Bewertung mehr möglich, wird die Verbindung von temporärem zu endgültigem Bewertungsdatensatz auf 0 gesetzt oder auf das korrespondierende Inserat abgeändert, für den Fall dass es sich hierbei um eine regelmäßige Fahrt mit einer noch

gültigen Buchung von Hin- bzw. Rückfahrt handelt (damit die laufende Bewertung auch fortgesetzt wird, anstatt eine Neue zu beginnen).

4.17 Seiten für die Übersicht vorgenommener bzw. ausstehender Bewertungen

- **Dateiname:**

- *MfzBewertungUebersicht.asp*
- *MfzBewertungUebersichtAbEr.asp* / *MfzBewertungAbErIns.asp* / *MfzBewertungAbErBuch.asp*
- *MfzBewertungUebersichtAbAb.asp* / *MfzBewertungAbAbIns.asp* / *MfzBewertungAbAbBuch.asp*
- *MfzBewertungUebersichtOffenFrei.asp* / *MfzBewertungOffenFreiIns.asp* / *MfzBewertungOffenFreiBuch.asp*
- *MfzBewertungUebersichtOffenNnFrei.asp* / *MfzBewertungOffenNnFreiIns.asp* / *MfzBewertungOffenNnFreiBuch.asp*
- *MfzBewertungUebersichtOffenGesperrt.asp* / *MfzBewertungOffenGesperrt.asp*

- **Zweck der Seite:**

Analog zur Buchungsübersicht wird auch hier ausgehend von der Hauptseite – in diesem Fall *MfzBewertungUebersicht.asp* – auf die entsprechenden Unterseiten zugegriffen. Außer von der Startseite bzw. den Linkleisten ist diese Übersicht auch über die Profilseite eines Nutzers erreichbar. Im Sinne der besseren Anschaulichkeit erfolgt eine Aufteilung in logische Abschnitte. Die bei den implementierten Dateien verwendeten Abkürzungen haben folgende Bedeutung:

- AbEr → abgeschlossene Bewertungen, die der \$REMOTE_USER von anderen Nutzern des Dienstes erhalten hat
- AbAb → abgeschlossene Bewertungen, die der \$REMOTE_USER selbst abgegeben hat
- OffenFrei → offene (d.h. noch nicht getätigte) Bewertungen, die zur Durchführung freigegeben sind
- OffenNnFrei → offene Bewertungen, die noch nicht freigegeben sind, weil die zugehörige Fahrt noch nicht stattgefunden hat
- OffenGesperrt → alle regelmäßigen Fahrten, die schon einmal bewertet wurden, sind ab der letzten Bewertung für eine Woche gesperrt, werden danach aber wieder freigegeben - weshalb sie auch in diesem Fall als offen eingeordnet sind

Bewertungen können vorgenommen, und zurückliegende Beurteilungen (erhaltene und abgegebene) im Detail betrachten werden.

▪ **Aufruf bzw. übergebene Parameter:**

(1) Selbstaufruf (Unterseiten)

→ Parameter:

Wie auch bei den anderen Übersichtsseiten wird der Selbstaufruf durch die Nutzung der Sortierfunktion bzw. der Tabellenaktivierung ausgelöst.

- `$pid_user` (abgeschlossene Bewertungen) → Die übergebene Nutzeridentifizierung (`$user1`) wird auf diese Variable abgebildet und über einen Selbstaufruf hinweg in der Session aufrechterhalten. Dieser Parameter wird jedoch nur bei der Anzeige der abgeschlossenen Bewertungen verwendet, da die restlichen Tabellen nur für den `$REMOTE_USER` bestimmt sind.
- `$sorterins/$sorterbuch`, `$ordererins/$ordererbuch`, `$richtungersins/$richtungerbuch` → Steuerung der Sortierung der Tabellen für erhaltene Bewertungen (getrennte Tabellen der Bewertungen als Inserent bzw. Buchender)
- `$sortabins/$sortabbuch`, `$orderabins/$orderabbuch`, `$richtungabins/$richtungabbuch` → Steuerung der Sortierung der Tabellen für abgegebene Bewertungen (getrennte Tabellen nach Inserent bzw. Buchender)
- `$sortofins/$sortofbuch`, `$orderofins/$orderofbuch`, `$richtungofins/$richtungofbuch` → Steuerung der Sortierung der Tabellen für die freigegebenen Bewertungsvorgänge stattgefundenen Fahrten (getrennte Tabellen nach Inserent bzw. Buchender)
- `$sortnnins/$sortnnbuch`, `$ordernnins/$ordernnbuch`, `$richtungnnins/$richtungnnbuch` → Steuerung der Sortierung der Tabellen für noch nicht freigegebene Bewertungsvorgänge (getrennte Tabellen nach Inserent bzw. Buchender)
- `$sortgsins/$sortgsbuch`, `$ordergsins/$ordergsbuch`, `$richtunggsins/$richtunggsbuch` → Steuerung der Sortierung der Tabellen für momentan gesperrte Bewertungsvorgänge (getrennte Tabellen nach Inserent bzw. Buchender)
- `$zeigeersins/$zeigeerbuch`, `$zeigeabins/$zeigeabbuch`, `$zeigeoffenfreiins/$zeigeoffenfreibuch`, `$zeigeoffennnfreiins/$zeigeoffennnfreibuch`, `$zeigeoffengesperrtins/$zeigeoffengesperrtbuch` → Zum Ein- und Ausblenden der Tabellen wird diese Variable als Statusmeldung der verantwortlichen Checkbox übergeben.

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

→ Parameter:

Dieser Aufruf führt zur Hauptübersicht, und kommt nur durch den `$REMOTE_USER` zustande, weshalb keine Nutzeridentifizierung übergeben werden muss. Wird jedoch eine der beiden Unterseiten der abgeschlossenen Bewer-

tungen angewählt, muss die ID des betreffenden Nutzers weitergeleitet werden.

- \$user1 (Unterseiten/abgeschlossene Bewertungen) → Diese Variable enthält die ID des Nutzers, dessen Bewertungen angezeigt werden sollen.

(3) Aufruf über die persönliche Profilseite

→ Parameter:

- \$user1 (Übersicht/abgeschlossene Bewertungen) → Diese Variable enthält die ID des Nutzers, dessen Bewertungen angezeigt werden sollen.

▪ Screenshot:

Bewertungsübersicht

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/ [Nach oben] [erhaltene Bewertungen] [abgegebene Bewertungen] [freigegebene offene Bewertungen]
[noch nicht freigegebene offene Bewertungen] [gesperrte offene Bewertungen]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:23 Uhr

Start Neues Angebot/Gesuch Meine Angebote/Gesuche Meine Buchungen Kontaktinfos

Bewertungen:

- eigener Bewertungsschnitt:
 - als Fahrer(in): Zuverlässigkeit: 4.2 / Auftreten: 4.2
 - als Mitfahrer(in): Zuverlässigkeit: 4.5 / Auftreten: 4.5

abgeschlossene Bewertungen

- 17 erhaltene Bewertung(en)
- 34 abgegebene Bewertungen

Hier können alle abgeschlossenen Bewertungen im Detail betrachtet werden

offene Bewertungen

- 6 freigegebene Bewertung(en)
- 0 noch nicht freigegebene Bewertung(en)
- 0 derzeit gesperrte Bewertung(en)

Hier finden Sie Details zu derzeit offenen Bewertungen bzw. zugehörigen Fahrten

Zurück

Abbildung 4-15 Screenshot Bewertungsübersicht (Hauptseite)

freigegebene offene Bewertungen

Hochschule Mittweida (FH)
University of Applied Sciences

Login: jbedlews/ [Nach oben]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:28 Uhr

Hinweis: Falls Sie für eine bestimmte Strecke Hin- und Rückfahrt gebucht haben, wird dies im Sinne der Bewertung als 'eine Fahrt' angesehen und kann deshalb nur einfach (statt doppelt) bewertet werden !

Start Neues Angebot/Gesuch Meine Angebote/Gesuche Meine Buchungen Meine Bewertungen

6 offene Bewertung(en) / freigegeben - eigene Angebote/Gesuche

Bewertung ab	von	Startort	Zielort	Typ	
So, 19. Sep 2010	F. Fischer	Mittweida	Flöha	Mitfahrer	
So, 19. Sep 2010	S. Weiß	Mittweida	Flöha	Mitfahrer	
So, 19. Sep 2010	F. Fischer	Flöha	Mittweida	Mitfahrer	
So, 19. Sep 2010	F. Fischer	Mittweida	Flöha	Mitfahrer	
Mi, 15. Sep 2010	S. Weiß	Flöha	München	Fahrer	
Mi, 15. Sep 2010	S. Weiß	Flöha	München	Fahrer	

0 offene Bewertung(en) / freigegeben - gebuchte Angebote/Gesuche

Zurück

Bewertung abgeben
Details des zugehörigen Angebotes/Gesuches betrachten

Abbildung 4-16 Screenshot Bewertungsübersicht (freigegeben)

[Start](#)
[Neues Angebot/Gesuch](#)
[Meine Angebote/Gesuche](#)
[Meine Buchungen](#)
[Meine Bewertungen](#)

☒ 5 abgegebene Bewertung(en) - als Inserent (J. Bedlewski)

Datum (Bewertung)	für	von	Zuverl.	Auftr.	
Mo, 13. Sep 2010	S. Weiß	J. Bedlewski	5	5	
Fr, 18. Jun 2010	S. Weiß	J. Bedlewski	5	5	
Sa, 6. Mrz 2010	S. Weiß	J. Bedlewski	2	2	
Do, 25. Feb 2010	S. Weiß	J. Bedlewski	4	4	
Do, 17. Sep 2009	S. Weiß	J. Bedlewski	5	5	

☒ 4 abgegebene Bewertung(en) - als Buchender (J. Bedlewski)

Datum (Bewertung)	an	von	Zuverl.	Auftr.	
Do, 22. Apr 2010	S. Weiß	J. Bedlewski	5	5	
Mo, 13. Sep 2010	S. Weiß	J. Bedlewski	5	5	
Mo, 13. Sep 2010	S. Weiß	J. Bedlewski	5	5	
Mo, 13. Sep 2010	S. Weiß	J. Bedlewski	3	3	

[Zurück](#)

[Details der Bewertung betrachten](#)

Abbildung 4-17 Screenshot Bewertungsübersicht (angegeben)

Funktionsweise:

(1) Selbstaufruf

Auch bei den hier verwendeten Übersichtsseiten erfolgt der Selbstaufruf durch Sortierung der Tabellen und deren Aktivierung bzw. Deaktivierung. Die verwendeten Mechanismen sind identisch mit den in vorherigen Abschnitten bereits besprochenen (siehe Seite 78/Seite 114).

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

Die Erzeugung der Tabellen und die entsprechende Verarbeitung der Daten unterscheidet sich nicht grundlegend von den anderen Übersichtsseiten des Dienstes. Bei den bereits getätigten Bewertungen existiert nur eine Detailansicht zur jeweiligen Bewertung selbst, weil der Datensatz des zugehörigen Inserats nicht mehr besteht. In den anderen Abschnitten kann noch auf die Inseratsdetails zugegriffen werden, und sobald eine Fahrt als freigegeben eingestuft ist, besteht auch Zugriff zum Formular.

Zu beachten ist jedoch, dass die Unterseiten der abgeschlossenen Bewertungen nicht ausschließlich vom betreffenden Nutzer selbst betrachtet werden können. Deshalb muss dort immer die Identität des Nutzers übergeben werden, dessen Daten angezeigt werden sollen. Dies trifft auch auf den vorliegenden Aufruf zu, obwohl hier nur der `$REMOTE_USER` in Frage kommt.

Auf allen Übersichtsseiten wird auch in diesem Fall eine Gültigkeitsprüfung des Aufrufs vorgenommen. Hierbei handelt es sich allerdings nur um die Abfrage nach Existenz und Gültigkeit des persönlichen Profils (`$REMOTE_USER` oder Fremd- ID – dies gilt also auch für Aufruf (3)).

(3) Aufruf über die persönliche Profilseite

Diese Übersichtsseite wird sowohl vom `$REMOTE_USER`, als auch den restlichen Nutzern verwendet. Jedoch sind nur die abgeschlossenen Bewertungen für alle Personen zugänglich. Entscheidend dabei ist die Variable `$pid_user`, welche bei Übereinstimmung mit dem `$REMOTE_USER` dazu führt, dass alle

Bewertungskategorien angezeigt werden, andernfalls nur die bereits abgeschlossenen (erhaltene und abgegebene). Die Variable `$showref` wird dafür entsprechend gesetzt → *all* für die Komplettanzeige – kein Inhalt für die eingeschränkte Ansicht. Der Zugriff auf die entsprechenden Unterseiten gestaltet sich identisch der in Punkt (2) beschriebenen Situation.

4.18 Seite zur Darstellung der Details einer Bewertung

- **Dateiname:**

- *MfzBewertungDetails.asp / MfzBewertung.asp*

- **Zweck der Seite:**

Auf dieser Seite kann der Nutzer Informationen zu einer erhaltenen (oder abgegebenen) Bewertung abrufen. Dabei werden die konkrete Bewertung selbst wiedergegeben, und einige Details zur Identifizierung der zugehörigen Fahrt dargestellt.

- **Aufruf bzw. übergebene Parameter:**

(1) Aufruf aus der Bewertungsübersicht

→ Parameter:

- \$mitf_bwid_fin → Dabei handelt es sich um die ID des zugehörigen permanenten Bewertungseintrags.

- **Screenshot:**



Abbildung 4-18 Screenshot Bewertungsdetails

- **Funktionsweise:**

(1) Aufruf aus der Bewertungsübersicht

Unter Nutzung des entsprechenden Schlüssels werden die darzustellenden Daten abgerufen und in den Quelltext eingefügt. Im ersten Teil der Seite sind einige Details zur Fahrt aufgeführt (Startort, Zielort, Datum, Frequenz, Inserent) und im zweiten Teil werden die konkreten Informationen zur Bewertung selbst wiedergegeben. Von dort aus gibt es (wie auch bei den Inseratsdetails) die Möglichkeit, auf das Nutzerprofil der an der Bewertung beteiligten Personen zuzugreifen.

Da es sich hierbei um eine Seite handelt, die jedem Nutzer zugänglich ist, muss nur geprüft werden, ob die übergebene ID stimmt. Dabei kommt es auch nur darauf an, dass ein passender Eintrag existiert.

4.19 Seite zum Verschicken einer Nachricht

- **Dateiname:**

- *formular.MfzMessage.asp / MfzMessageLib.asp*

- **Zweck der Seite:**

Über dieses Formular hat der Nutzer die Möglichkeit ohne den Einsatz eines separaten Programms zum Versand von E-Mails einem anderen Nutzer des Mitfahrdienstes eine einfache Nachricht zukommen zu lassen.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- `$pid_user` → Der übergebene Login des Nutzers, dem die Nachricht geschickt werden soll, wird auf diese Variable abgebildet. Wird ein Selbstaufruf ausgelöst, wird diese ID zur weiteren Verarbeitung aufrechterhalten.

- `$zeigemessage` → Auch wenn dieses Formular nur eine Funktionskomponente enthält, kann dieses Textfeld wie die Tabellen der einzelnen Übersichtsseiten an- und ausgeschaltet werden. Das ist in der Einheitlichkeit des Designs der Seiten begründet, die nicht als Hauptkomponenten eingestuft sind (es wurde sich hierbei am Wohnungsmarkt orientiert). Diese Variable übergibt den jeweiligen Zustand.

- (2) Aufruf aus dem persönlichen Profil

- Parameter:

- `$user1` → Diese Variable enthält die ID des Nutzers, dem die Nachricht geschickt werden soll.

- **Screenshot:**

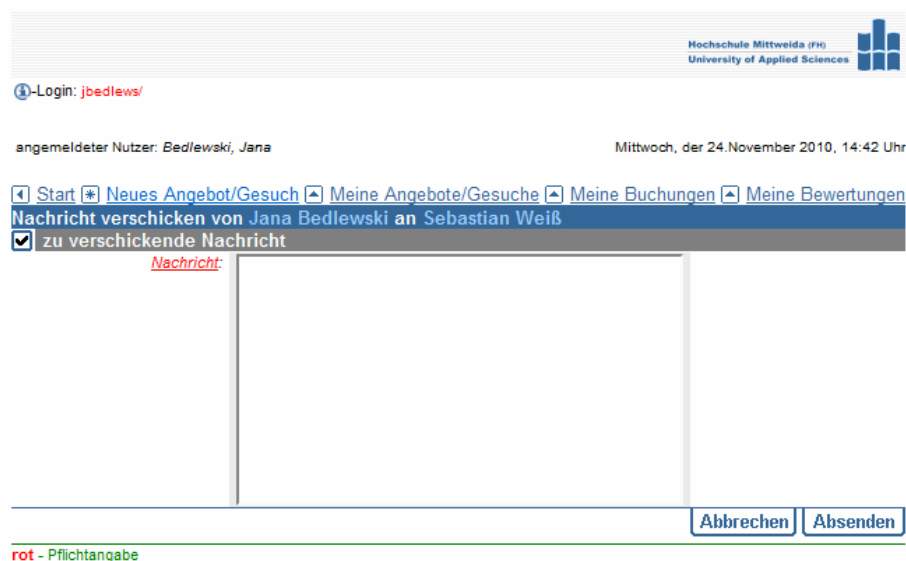


Abbildung 4-19 Screenshot Mailformular

- **Funktionsweise:**

- (1) Selbstaufruf

Wird das Textfeld aktiviert bzw. deaktiviert oder schickt der Nutzer das Formular mit einem leeren Textfeld ab, wird der Selbstaufruf ausgelöst. Bei letzterem reagiert die Eingabeprüfung, da ein Absenden nur Sinn macht, wenn auch eine Mitteilung eingegeben wurde.

- (2) Aufruf aus dem persönlichen Profil

Die Identifizierung des Nutzers, an den die Nachricht geschickt werden soll, wird bei erfolgreicher Eingabe an die Mailfunktion

`mail_message($pid_user)` weitergegeben. Dort wird die E-Mail erstellt, das heißt alle Teile wie Sender, Empfänger, Betreff und der Text werden zusammengefügt um danach unter Nutzung der Standard PHP- Funktion

`mail()` ohne Einsatz eines eigenständigen Programms dieser Art eine E-Mail an den betreffenden Nutzer zu senden. Außerdem wird mit Hilfe der Variable `$pid_user` die Weiterleitung zurück zur passenden Profilseite realisiert (Weiterleitung nach dem Absenden/Abbruch der Eingabe), sowie die Anzeige der Titelleiste des Formulars auf den derzeitigen Nutzer angepasst. Dort kann nochmals kontrolliert werden, wer welchem Nutzer die Nachricht schickt.

Zur Sicherheit wird beim Aufruf der Seite geprüft ob Sender und Empfänger ein (gültiges) Profil besitzen. Trifft dies auf den Empfänger nicht zu, wird die entsprechende Variable automatisch auf den `$REMOTE_USER` gesetzt.

4.20 Seite zur Eintragung des persönlichen Profils

- **Dateiname:**

- `formular.MfzProfil.asp / MfzProfilPerson.asp`

- **Zweck der Seite:**

Zur vollständigen Nutzung der Funktionen des Mitfahrdienstes muss ein Nutzer einige wichtige persönliche Angaben tätigen, was über dieses Formular geschieht. Außer den Mindestangaben können auch noch weitergehende Informationen zur Beschreibung der eigenen Person eingetragen werden, welche dann auch teilweise bei der Suche nach Inseraten als Unterscheidungskriterien hinzugezogen werden können. Im Allgemeinen werden diese Daten auf einer Profilseite zusammengefasst dargestellt, um die jeweilige Person anderen Nutzern vorzustellen, und so ein besseres Bild des- oder derjenigen zu ermöglichen.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- `$R0` → Um die Pflichtangabe zur Kontaktdatenveröffentlichung in der Eingabeprüfung berücksichtigen zu können, muss die betreffende Variable separat übergeben werden (da es sich hierbei nicht um eine objektbasierte Komponente handelt).

- `$zeigeinf, $zeigepers, $zeigefahrt, $zeigeprofil` → Beim Ein- und Ausblenden der einzelnen Formularabschnitte werden diese Variablen mit der jeweiligen Statusinformation im Zuge eines Selbstaufrufs übergeben

- (2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

- Parameter:

- Da dieses Formular immer nur für den jeweiligen `$REMOTE_USER` zugänglich ist, muss zur Identifizierung bzw. Unterscheidung kein Login übergeben werden.

- Wird dieses Formular über die Linkleiste aufgerufen, ist es jedoch nötig, folgenden Parameter zu übergeben (die Startseite ist als Standardwert festgelegt):

- `$back` → beim Abbruch der Eingabe wird durch diesen Parameter sichergestellt, dass der Nutzer zu der Seite zurückkehrt, von der er zu diesem Formular gelangt ist

- **Screenshot:**

persönliches Profil bearbeiten Hochschule Mittweida (FH) University of Applied Sciences

Login: jbedlews/ [Nach oben]

angemeldeter Nutzer: Bedlewski, Jana Mittwoch, der 24. November 2010, 14:42 Uhr

Start | neues Angebot/Gesuch | Meine Angebote/Gesuche | Meine Buchungen | Meine Bewertungen

persönliches Profil

☒ **Kontaktinfos**

Vorname: Jana
 Name: Bedlewski
 Seminargruppe: CT04w1
 E-Mail: jbedlews@htwm.de
 Festnetznummer: ☐
 Mobilfunknummer: ☒
 Kontaktdaten veröffentlichen?: ☒ ja ☐ nein

☒ **persönliche Einstellungen**

Raucher?: ☐ ja ☒ nein
 m/w?: ☐ m ☒ w
 Sprachen?: ☒ deutsch ☒ englisch ☐ spanisch ☐ italienisch ☐ französisch ☐ russisch ☐ andere

☒ **Fahrtseinstellungen**

Raucher: unerwünscht
 m/w: egal
 Gepäckmitnahme: Handgepäck
 Tiere: egal

☒ **Profileinstellungen**

Profil deaktivieren?: ☐ deaktivieren

Abbrechen Prüfen Speichern

☒ korrekte Eingabe prüfen
 'VHS' - Verhandlungssache
 rot - Pflichtangabe
 blau - freiwillige Angabe

Abbildung 4-20 Screenshot Formular persönliches Profil

- **Funktionsweise:**

(1) Selbstaufruf

Ein Selbstaufruf wird ausgelöst durch die Aktivierung der Segmente, sowie bei der separaten Eingabeprüfung und einer negativen Prüfung während des Speichervorgangs. Diese bezieht sich auf die Pflichteingaben der wichtigsten Details, sowie die schematisch korrekte Eingabe der Mailadresse und der Telefonnummern. In den global verfügbaren Bibliotheken existieren dazu Routinen zur Überprüfung der grundlegend gültigen Struktur verschiedener Datentypen. Bei der Mailadresse wird dabei eine Funktion auf den Eingabestring zugewiesen:

```
...
    $mail_check = string_isEmail($ad_email);
...

```

Für die Telefonnummern existiert eine eigene Klasse (*FormObjectTel.asp*), welche bei der Definition des entsprechenden Objektes in der Datei *MfzProfilPerson.asp* angegeben werden muss.

Eine weitere Pflichtangabe ist die Festlegung, ob die Kontaktdaten auf der Profilseite für alle Nutzer jederzeit frei einsehbar sein sollen. Diese Auswahl erfolgt mittels Radiobutton.

(2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

Neben der speziellen Eingabeprüfung unterscheidet sich dieses Formular nicht wesentlich von anderen Seiten dieser Art in der Handhabung und Funktionalität. Die Aufnahme der getätigten Angaben in die Datenbank erfolgt über die objektbasierte Speicherfunktion. Für ein eventuelles Update wird ebenso nur die Variable `$REMOTE_USER` benötigt. Danach werden die einzelnen Elemente des Formulars mit den bereits vorhandenen Informationen belegt. Dabei ist für die Radiobuttons (Kontaktdaten, Raucher, Geschlecht) eine eigenständige Zuweisung notwendig, aufgrund der Tatsache, dass Radiobuttons zum Zeitpunkt der Bearbeitung dieser Programmierung noch nicht Teil der objektbasierten Komponenten waren (und demnach auch keine entsprechende Definition zur einfachen Implementierung existierte).

Eine weitere wichtige Komponente dieses Formulars ist die Möglichkeit, das persönliche Profil deaktivieren zu können. Damit würde sich die Handhabung des Dienstes genauso darstellen, als ob noch kein Profil eingetragen ist. Jedoch soll es (derzeit) keine Möglichkeit für den Nutzer selbst geben, sein Profil komplett löschen zu können. Beispielsweise wäre es ja so möglich, bei schlechten Bewertungen einfach die Person zu löschen, um dann „sauber“ ein neues Profil aufsetzen zu können. Jedoch wird der Zeitpunkt der Deaktivierung abgespeichert und in Zukunft könnte man eine Funktion einbauen, die nach einem gewissen Zeitraum der Inaktivität das Profil löscht. Nach der Deaktivierung bleiben im Grunde nur die abgeschlossenen Bewertungen bestehen - alle Inserate und Buchungen des Nutzers werden (mit geringfügigen Anpassungen) entsprechend des üblichen Löschvorganges dieser Eintragungen gehandhabt. Das heißt, die Einträge werden gelöscht, so lange diese nicht mehr für eine Bewertung benötigt werden (ansonsten wird Status auf 1 gesetzt). Im Gegensatz zur standardmäßigen Löschroutine wird der zugehörige Bewertungsvorgang für den zu deaktivierenden Nutzer allerdings gesperrt (*bew_block* auf 1 und *next_bew* auf 0).

4.21 Seite zur Eintragung des Fahrzeugprofils

- **Dateiname:**

- `formular.MfzProfilKfz.asp / MfzProfilKfz.asp`

- **Zweck der Seite:**

Möchte ein Nutzer über diesen Mitfahrdienst Sitzplätze für eine Fahrt zur Verfügung stellen, wählen einige beschreibende Details zum Fahrzeug selbst durchaus sinnvoll, etwa zum schnelleren Auffinden eines in den meisten Fällen unbekannten Fahrzeugs am Treffpunkt. Mit diesem Formular bietet sich ein fester Anlaufpunkt zur zentralen Verwaltung dieser Daten, welche letztendlich (wenn freigegeben) gemeinsam mit den Informationen zur Person auf der eigenen Profilseite veröffentlicht werden.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- `$R1` → Diese Variable sagt aus, in welchem Zustand sich die beiden Radiobuttons befinden, welche anzeigen sollen, ob der betreffende Nutzer seine Fahrzeugdaten veröffentlichen will.
 - `$zeigeKFZ` → Dieses Formular besteht zwar nur aus dem Segment „Fahrzeugdetails“, aber um das einheitliche Design beizubehalten, kann auch dieser Abschnitt an- und ausgeschaltet werden.

- (2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

- Parameter:

- Analog zum Formular für persönliche Details, muss beim Aufruf dieser Seite keine Nutzeridentifizierung übergeben werden. Lediglich der Referenzparameter ist auch hier notwendig, um zur vorherigen Seite zurückzukehren.

- `$back` → Wird die Eingabe abgebrochen, kehrt der Nutzer zu der Seite zurück, auf welche in dieser Variable verwiesen wird.

- **Screenshot:**

Abbildung 4-21 Screenshot Formular Fahrzeugprofil

- **Funktionsweise:**

- (1) Selbstaufruf

Im Zuge der selbstständigen oder der automatisch ablaufenden Eingabeprüfung vor der endgültigen Speicherung wird ein Selbstaufruf ausgelöst. Genauso beim Aus- und Anschalten des vorhandenen Segmentes. Es erfolgt jeweils eine simple Überprüfung der übergebenen Variablen. Die Eingabeprüfung der Parameter beschränkt sich hier auf die korrekte Anzahl der standardmäßig freien Sitzplätze (mindestens 1/höchstens 9 → optionaler Parameter), und die Pflichtangabe zwecks Veröffentlichung dieser Profildaten. Genau wie bei den persönlichen Angaben eines Nutzers sind diese Informationen entweder für alle Personen zugänglich (ja), oder nur die Personen, die ein Inserat bei dem betreffenden Nutzer gebucht haben (nein).

- (2) Aufruf über die Startseite/die Linkleiste auf einzelnen Dienstseiten

Wie auch bei dem persönlichen Profil wird während des Aufrufs dieser Seite für den gerade aktiven `$REMOTE_USER` eine Abfrage der entsprechenden Tabelle der Datenbank durchgeführt. Ist noch kein Eintrag dieser Art vorhanden, wird das Formular mit leeren Elementen initialisiert. Andernfalls wird der entsprechende Schlüssel ausgelesen, dem Datenbankobjekt übergeben wodurch automatisch alle relevanten Daten ausgelesen und den zugehörigen Elementen zugewiesen werden (der Inhalt der Radiobuttons muss jedoch auch hier eigenständig zugeordnet werden). Die Verarbeitung selbst gestaltet sich identisch zum persönlichen Profil.

Sollte der Nutzer noch kein Profil eingetragen oder dieses deaktiviert haben, ist der Aufruf ungültig und es erfolgt die Weiterleitung zur Startseite.

4.22 Seite zur Darstellung der personenbezogenen Informationen

- **Dateiname:**

- *MfzPersonInfo.asp*

- **Zweck der Seite:**

Auf dieser Seite werden alle Angaben, die der Nutzer über das persönliche und das Fahrzeugprofil getätigt hat, an einem Ort übersichtlich zusammengefasst. Zusätzlich dazu besteht von hier aus ein direkter Zugriff auf die Bewertungen der betreffenden Person, sowie dessen derzeit gültige Angebote und Gesuche.

- **Aufruf bzw. übergebene Parameter:**

- (1) Selbstaufruf

- Parameter:

- `$zeigeinf`, `$zeigebew`, `$zeigefahropt`, `$zeigefahrinf`, `$zeigefahrins` → Diese Detailseite ist in mehrere Kategorien unterteilt, welche Aktiv bzw. Inaktiv geschaltet werden können, was jeweils in einem Selbstaufruf resultiert. Die hier genannten Variablen übergeben dabei den aktuellen Zustand.
 - `$pid_user` → Bei übergebenem Login wird selbiger auf diese Variable zugewiesen. Bei Aufruf von der Startseite wird dabei der `$REMOTE_USER` eingesetzt.

- (2) Aufruf über die Startseite/die verschiedenen Detailseiten/die Suchergebnisliste (Personensuche)

- Parameter:

- `$user1`(nicht bei Startseite) → Anhand dieses Parameters wird festgelegt, wessen Daten auf dieser Seite dargestellt werden. Da der Aufruf von der Startseite nur vom `$REMOTE_USER` durchgeführt wird, muss die Identifizierung nicht extra übergeben werden.

- **Screenshot:**

detaillierte Personeninfos Hochschule Mittweida (FH) University of Applied Sciences

Login: [jbedlews/](#) [[Nach oben](#)]

angemeldeter Nutzer: *Bedlewski, Jana* Mittwoch, der 24. November 2010, 14:44 Uhr

[Start](#)
[Meine Buchungen](#)
[Kontaktinfos](#)
[Fahrzeuginfos](#)

Nutzerprofil

☒ **persönliche Informationen**

Name: **Jana Bedlewski**
 Seminargruppe: **CT04w1**
 E-Mail: jbedlews@htwm.de [Nachricht schicken](#)
 Festnetznummer: (keine Angabe)
 Mobilfunknummer: (keine Angabe)

Raucher?: **nein**
 m/w?: **weiblich**
 Sprachen?: **deu eng**

☒ **Bewertungen**

als Fahrer(in): Zuverlässigkeit: 4.2 Auftreten: 4.2
 als Mitfahrer(in): Zuverlässigkeit: 4.5 Auftreten: 4.5
 Bewertungsdetails: [Details betrachten](#)

☒ **bevorzugte Fahrtoptionen**

Raucher?: **unerwünscht**
 m/w?: **egal**
 Gepäck?: **Handgepäck**
 Tiere?: **egal**

☒ **Fahrzeuginformationen**

Kennzeichen: **FG-123**
 Automarke: **Opel**
 Autotyp: **Kadett**
 Autofarbe: **rot**
 Kommentar: (keine Angabe)

☒ **eingetragene Angebote/ Gesuche**

Angebote: **3** [eingetragene\(s\) Angebot\(e\)](#)
 Gesuche: **1** [eingetragene\(s\) Gesuch\(e\)](#)

[Zurück](#)

Abbildung 4-22 Screenshot Profildetails

- **Funktionsweise:**

(1) Selbstaufruf

Nach demselben Schema wie etwa bei den Formularen zur Vervollständigung der Profilinformationen wird hier ein Selbstaufruf ausgelöst, wenn einer der Abschnitte aus- bzw. später wieder eingeblendet wird. Weitere Szenarien für einen Selbstaufruf sind bei dieser Seite nicht vorgesehen, da weder eine Fehlerprüfung vorgesehen ist (Seite ist kein Formular), noch aktive Komponenten implementiert sind.

(2) Aufruf über die Startseite/die verschiedenen Detailseiten/die Suchergebnisliste (Personensuche)

Unter Nutzung des zugrunde liegenden Logins werden die benötigten persönlichen und Kfz-Daten abgerufen und in Textform in die zugehörigen Abschnitte der Profilübersicht eingefügt. Sollte es der betreffende Nutzer vorgezogen haben, bestimmte Angaben offen zu lassen, dann werden diese mit „keine Angabe“ gekennzeichnet, anstatt die Spalten einfach auszublenden.

Eine der wichtigsten Komponenten dieser Seite ist die Kontaktmöglichkeit zum betreffenden Nutzer. Bei entsprechender Freigabe wird die E-Mail Adresse eingeblendet (kann zum Verschicken über ein separates Mailprogramm

genutzt werden → über `mailto:`). Aber in jedem Fall besteht von dieser Übersicht aus Zugang zum Mailformular, womit direkt von der Seite aus Nachrichten per E-Mail versandt werden können (auf Basis einer standardmäßigen PHP Funktion).

Im Abschnitt für die Bewertungen führt ein Link zur konkreten Bewertungsübersicht, aber für den ersten Überblick wird zusammenfassend der Durchschnitt für Auftreten und Zuverlässigkeit getrennt nach Fahrer bzw. Mitfahrer aufgelistet. Dabei wird der jeweils ausgelesene Wert für die bessere Optik noch gerundet und auf eine Nachkommastelle begrenzt:

...

```
$zu_mit=round($row_pid_user["zu_mit"], 1);  
$auf_mit=round($row_pid_user["auf_mit"], 1);
```

...

Die hohe Anzahl selbst erstellter Label ist bei dieser Seite damit zu erklären, dass im Rahmen des angewandten Designs (für den Bereich Angaben zur Person) hier bei der Detailseite anstatt der standardmäßig in blau gerenderten Beschriftung ein grauer Farbton erscheinen soll.

4.23 Seite zur Durchsuchung eingetragener Nutzerprofile

- **Dateiname:**

- *index.asp / MfzSuche.asp*

- **Zweck der Seite:**

Mittels dieser Personensuche ist es möglich, die Infoseite eines bestimmten Nutzers direkt aufzurufen, ohne erst über ein Inserat, eine Buchung oder eine Bewertung dieser Person gehen zu müssen.

- **Screenshot:**

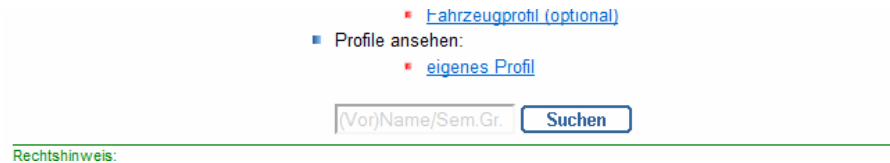


Abbildung 4-23 Screenshot Personensuche

- **Aufruf bzw. übergebene Parameter:**

(1) Einfache Suche → direkt auf der Startseite

→ Parameter:

Da es sich hierbei um den Standardaufruf der Startseite handelt, werden keine speziellen Parameter für die Suche übergeben.

- **Funktionsweise:**

(1) Einfache Suche → direkt auf der Startseite

Diese Suchvariante beschränkt sich auf einen Parameter. Das heißt, die Texteingabe kann dazu genutzt werden, entweder Vor- und/oder Nachname einer Person, oder die Seminargruppe anzugeben. Wird mir einem kompletten Namen gesucht (das bedeutet in diesem Fall zweiteilig), bezieht sich Teil 1 auf den Vornamen und Teil 2 auf den Nachnamen. Bei der Eingabe nur eines Teils wird jedoch nach Übereinstimmungen in beiden Abschnitten gesucht.

Es sollte auch in diesem Zusammenhang erwähnt werden, dass sich die Suche auf die (teils optionalen) Eintragungen im persönlichen Profil bezieht. Fehlerhafte oder fehlende Eintragungen können deshalb die Suche (ungewollt) verfälschen.

Nach Bestätigung der Eingabe wird der Nutzer auf die Seite *MfzPersonSucheErgebnisUebersicht.asp* weitergeleitet, wo nach dem Vorbild sämtlicher Übersichtsseiten des Dienstes die gefilterte Nutzerliste zu finden ist (nach entsprechend verarbeiteter Eingabe).

4.24 Seiten zur Verarbeitung und Darstellung der Suchergebnisse

- **Dateiname:**

- *MfzPersonSucheErgebnisUebersicht.asp / MfzPersonSucheErgebnis.asp*

- **Zweck der Seite:**

Die auf der Startseite im Suchfeld für Personen getroffene Texteingabe wird auf dieser Seite verarbeitet und das Resultat dieser Anfrage wird anschließend in einer Übersichtstabelle aufgelistet.

- **Aufruf bzw. übergebene Parameter:**

(1) Selbstaufruf

→ Parameter:

Zur Steuerung der Sortierung und korrekten Darstellung der Tabellen werden folgende Parameter übergeben:

- \$sortpers, \$orderpers, \$richtungpers → Diese Parameter steuern die Sortierung der Ergebnisliste (durch Beeinflussung der Datenbankabfrage).
- \$zeigebers → Der Status der Checkbox beim Aus- und Einblenden wird in dieser Variable gespeichert.

Genau wie bei der Suche für Inserate, wird der von der Startseite übergebene Suchparameter in einer Session zwischengespeichert, um nach einem Sortierungsvorgang (allgemein einem Selbstaufruf) die Ergebnisliste aufrechterhalten zu können.

(2) Aufruf durch Weiterleitung von der Startseite

→ Parameter:

Nach der Bestätigung der Suche auf der Startseite wird der Suchparameter \$person_einfach vom Formularabschnitt übergeben, und auf dieser Seite in die Datenbankabfrage eingefügt.

- **Screenshot:**

The screenshot shows a web application interface. At the top right is the logo of Hochschule Mittweida (FH) University of Applied Sciences. Below it, a login bar shows 'Login: jbedlews/'. Further down, it says 'angemeldeter Nutzer: Bedlewski, Jana' and 'Mittwoch, der 24. November 2010, 14:53 Uhr'. A navigation bar contains links: 'Start', 'Neues Angebot/Gesuch', 'Meine Angebote/Gesuche', 'Meine Bewertungen', and 'Kontaktinfos'. Below this, a checkbox '2 Person(en) gefunden' is checked. A table displays the search results:

Name	Sem.Gr.	m/w ?	Raucher ?
Jana Bedlewski	CT04w1	w	nein
Sebastian Weiß	CT04w1	m	nein

At the bottom right of the table area is a 'Zurück' button. At the bottom left, there is a link 'persönliches Profil betrachten'.

Abbildung 4-24 Screenshot Suchergebnisse Personensuche

- **Funktionsweise:**

- (1) Selbstaufruf

Die grundlegende Handhabung ist identisch mit der Suche für Inserate. Im Gegensatz dazu wird hier jedoch eine wesentlich geringere Zahl an Parametern verarbeitet (nur ein Such- und ein Aktivierungsparameter).

- (2) Aufruf durch Weiterleitung von der Startseite

Im Vergleich zur Inseratssuche ist die grundlegende Struktur des Suchalgorithmus wieder zu erkennen. Da aber nur ein Parameter verarbeitet wird und auch keine Trennung der Abfrage nötig ist wie bei Angeboten und Gesuchen, gestaltet sich die Datenbankabfrage wesentlich simpler.

Vor- und Nachname werden bei dieser reinen Personen- genau wie bei der Inseratssuche in einem einzelnen Textfeld angegeben. Deshalb müssen beide Teile vor der Verwendung in der Abfrage voneinander getrennt werden:

...

```
$array= explode(" ", $person_einfach);  
$vn = $array[0];  
$nn = $array[1];
```

...

Hat der Nutzer die Suche ohne Texteingabe gestartet, wird die konkrete Verarbeitung der ermittelten Variablen umgangen. Ansonsten wird bei der SQL-Abfrage unterschieden, ob beide Variablen belegt sind, oder nur ein Teil eines Namens bzw. eine Seminargruppe angegeben wurde. Der folgende Abschnitt zeigt den Hauptteil des resultierenden Abrufs für den letzteren Fall:

...

```
$db_suche_pers.=" WHERE (((name_vn LIKE '%$vn%') OR (name_nn  
LIKE '%$vn%')) OR semgr LIKE '%$vn%') AND status!='1' ORDER  
BY $_SESSION[orderpers] $_SESSION[richtungpers]";
```

...

Dabei müssen die beiden Variablen `$vn` und `$nn` nicht genau mit den entsprechenden Parametern des Datensatzes übereinstimmen. Stattdessen kommt eine Mustersuche zur Anwendung, wobei der enthaltene String nur im Vergleichsstring enthalten sein muss. Dafür werden sog. „Wildcards“ verwendet (in diesem Fall das Prozentzeichen), welche es ermöglichen, dass sich sowohl vor, als auch nach dem Suchstring beliebige Zeichen in beliebiger Menge befinden können. Das „OR“ in der Abfrage stellt dabei kein exklusives, sondern ein nichtausschließendes oder dar, was bedeutet, dass mindestens eine der verknüpften Bedingungen zutreffen muss (es können aber auch mehrere oder alle sein).

5 Zusammenfassung und Ausblick

5.1 Ergebnisse

Die Aufgabe des in dieser Bachelorarbeit behandelten Projektes war es, für das Studentenportal der Hochschulwebsite einen Dienst zur Vermittlung von Mitfahrgelegenheiten zu entwerfen und praktisch umzusetzen.

Um eine Basis für den Dienstentwurf zu schaffen, wurden zuerst einige im Internet vertretene Dienste dieser Art auf ihre Bestandteile hin analysiert. Eine bereits im Jahr 2003 für die Hochschulwebsite realisierte Version eines Mitfahrdienstes diente dabei als zweite wichtige Informationsgrundlage. Letztendlich ergab sich daraus unter Einbeziehung der Voraussetzungen des Studentenportals (und der damit verbundenen Zielgruppe) ein genaues Bild des Dienstes. Dieses wurde dann im abschließenden Teil der theoretischen Betrachtungen in ein konkretes Konzept umgesetzt, worin alle zu realisierenden Funktionsbereiche im Detail beschrieben sind.

Im praktischen Teil dieser Arbeit wurden die erforderlichen Komponenten dann auf Grundlage der Programmiersprache PHP erstellt, wobei JavaScript an einigen Stellen hilfreich zum Einsatz kam. Die Verwaltung der Daten erfolgt dabei unter Nutzung einer MySQL Datenbank, auf die mittels SQL Befehlen und teilweise unter Nutzung von Datenbankobjekten zugegriffen wird. Die Funktionalität des Dienstes konnte so mit insgesamt 67 selbst erstellten Dateien (dynamische Seiten und Bibliotheken) erreicht werden. Die dazugehörigen Quelltexte als Resultat des praktischen Teils sind im Anhang beigelegt.

Mit der Fertigstellung dieses Projektes wurde eine grundlegende Realisierung des Konzeptes „Vermittlung von Mitfahrgelegenheiten über das Studentenportal der Hochschulwebsite“ erreicht. Die wichtigsten Funktionsgruppen sind in einer ersten Form erarbeitet und funktionsfähig implementiert worden, wodurch ein einsatzfähiger Dienst vorliegt. Für die eventuelle spätere Weiterentwicklung und Verbesserung dieser Umsetzung sind in Kapitel 5.2 bereits einige mögliche Optionen angeführt.

5.2 Erweiterungsvorschläge

Um das allgemeine Konzept eines Mitfahrdienstes grundlegend zur Funktion bringen zu können, benötigt es nicht viele Komponenten, und innerhalb dieses Projektes wurden wichtige Bestandteile in einer ersten Form umgesetzt. Jedoch gibt es auch immer Raum für Erweiterungen und Verbesserungen, und die folgende Liste soll dafür einige mögliche Optionen aufzeigen.

- **allgemeine Anmerkungen**

- Bessere Anbindung zu bestehenden Strukturen der Hochschule. So könnte beispielsweise die Handhabung der Termine (bei Suche, Fahrten-/Buchungsmanagement etc.) mit den Stundenplänen der Seminargruppen oder dem allgemeinen Terminmanagement verbunden werden.
- Obwohl der Dienst primär für Fahrten zur Hochschule und zurück genutzt werden wird, ist es zu überlegen, ob ein gesondert gekennzeichnete Abschnitt für reine (einmalige) Urlaubs- bzw. Freizeitfahrten sinnvoll wäre. So könnten sich dann ja beispielsweise auch wie bei einer WG nette Gruppen zusammenfinden (und Geld sparen!).
- Möglicherweise könnte der Dienst auch auf andere Bereiche im Sinne der Mobilität erweitert werden, wie etwa die Vermittlung von Bahnreisenden (siehe Wochenendticket/Sachsenticket etc.), oder auch Transportmöglichkeiten.
- Als kleine hilfreiche Zusatzfunktion für den Parameter Preis ließe sich eine Art Rechner integrieren, wie er bei diversen Dienstangeboten im Internet zu finden ist. Der Nutzer müsste Spritpreis und Kilometerzahl grob abschätzen, und würde so einen gewissen Richtwert erhalten.
- Für das Profil des eigenen Fahrzeugs wäre durchaus eine Option sinnvoll, zusätzlich zur Beschreibung ein Foto hochladen zu können. Bei unbekannten Fahrern ergäbe sich hierdurch ein wesentlich höherer Erkennungswert.
- Die Angaben im persönlichen Profil sind zwar Pflicht, aber derzeit völlig frei wählbar. Eine teilweise Überprüfung wäre im Sinne der Seriosität des Dienstes durchaus sinnvoll. Name und Vorname könnten aus den vorliegenden Daten der Hochschule entnommen werden (wie es in Version von Sandro Gerstenberger schon der Fall war); was den äußerst wichtigen Parameter der E-Mail Adresse betrifft, wäre eine Art Prüfung denkbar, bei welcher zuerst eine Nachricht an die angegebene Adresse geschickt wird, und der Nutzer dann einem vorgegebenen Link folgen muss, um die Adresse (und damit das Profil) zu bestätigen.
- Eine Art Routenplaner wäre mit Sicherheit zu aufwendig (und in diesem Fall auch nur bedingt sinnvoll), aber zumindest eine Einbindung von Google Maps könnte sich als praktikabel erweisen. Einerseits um die entsprechenden Start-/Zielorte auf einer Karte anzeigen zu können

(eine Erweiterung auf optionale Adresseingabe wäre hier sinnvoll). Andererseits könnte die gesamte Strecke auf der Karte dargestellt werden (auch unter Einbeziehung selbstdefinierter Zwischenhaltepunkte). In diesem Fall wären mögliche Ausweichoptionen für Start- und/oder Zielort direkter erkennbar.

- Mit der immer stärker aufkommenden Beliebtheit und Verbreitung von so genannten „Apps“ wäre eine eigenständige Anwendung in dieser Richtung sicher sinnvoll.

▪ **Inserate / Buchungen / Bewertungen**

- Abgesehen vom bereits angesprochenen Terminmanagement wäre (wie auch schon in der Arbeit von Sandro Gerstenberger erwähnt) eine zeitliche Differenzierung der Angebote und Gesuche sinnvoll - 1./2. Woche bzw. durchgehend.
- Die Platzverwaltung könnte präzisiert werden, indem auch die Zwischenhalte einbezogen/unterschieden werden (anstatt eine Buchung was die Plätze angeht immer für die ganze Fahrt zu zählen).
- Nach dem Vorbild anderer Dienste wäre es denkbar, eine Art Inseratsfreigabefunktion einzuführen. Das heißt, nach dem ein Angebot oder Gesuch eingetragen wurde, muss der Inserent den Eintrag noch freigeben. In dem Sinne könnten Inserate im Laufe ihrer Zeit bei Bedarf aktiviert/deaktiviert werden, ohne diese jedes Mal löschen bzw. neu eintragen zu müssen. Eine entsprechende farbliche Kennzeichnung in der Übersichtsseite wäre in dem Fall angebracht.
- Inserate könnten von vorne herein mit einer Art „Verfallsdatum“ versehen sein, um sog. „Inseratsleichen“ zu verhindern (regelmäßige Einträge die eigentlich nicht mehr angeboten werden, aber vom Inserenten nicht gelöscht wurden). In dem Fall müsste natürlich auch eine Verlängerungsmöglichkeit integriert werden.
- Für die Start- und Zielorte könnte es eine eigene Tabelle in der Datenbank geben, um etwa bei Eingabe eines (oder mehrerer) Buchstaben passende Vorschläge zu liefern (in Form einer aufklappenden Liste), oder eine direkte Auswahl aus einer Liste zu ermöglichen. Da es sich hierbei eher um einen Dienst mit kleinerem Umfang handelt der nicht „professionell“ administriert wird, wäre eine selbstlernende Tabelle sicher die bessere Lösung (anstatt einer per Hand erstellten festen Datenbasis).

▪ **Nachrichten verschicken**

- Die Funktion zum Versenden von Mitteilungen von der reinen Verwendung im Profil lösen. So wäre es denkbar, in ausgewählten Übersichtsseiten einen entsprechenden Funktionsbutton anzubieten (ähnlich Änderung eines Inserats, Aufruf der Detailseite etc.). Dabei wäre auch eine Option wie „Gruppenmail“ an bestimmten Punkten durchaus sinnvoll, etwa um eine Mitteilung an alle Buchenden bzw. alle Buchenden eines bestimmten Inserats zu verschicken.

-
- Wird die Nachrichtenfunktion im Zusammenhang mit bestimmten Einträgen verwendet (Inserate, Buchungen oder Bewertungen), könnte es die Auswahloption geben, entsprechende Details (zur besseren Unterscheidung) in die Mail einzufügen.
 - Im persönlichen Profil könnte es eine Einstellung geben, mit welcher der Nutzer steuern kann, ob die verschiedenen Benachrichtigungsmails akzeptiert werden oder nicht (sollte sich nicht auf die normalen Nachrichten beziehen, da eine grundlegende Kommunikation immer möglich sein muss).
 - Die Benachrichtigungen könnten in dem Sinne erweitert werden, dass die Nutzer automatisch an wichtige Dinge bzw. Zeitpunkte per Mail erinnert werden (anstehende Fahrt, Bewertung etc.), wobei die entsprechende Zeitspanne im Profil festgelegt werden könnte.
- **Programmierung**
 - Insgesamt könnte die programmtechnische Umsetzung noch konsequenter auf die Verwendung der objektbasierten Programmierung eingestellt werden. Dazu gehören Abschnitte, für die zum jetzigen Stand teilweise bereits Funktionen existieren (siehe Datenbankabfragen). In anderen Bereichen fehlen noch entsprechende Definitionen bzw. sind noch nicht in den Standardbibliotheken eingearbeitet. Eine Komponente wie der Radiobutton kann beispielsweise im Gegensatz zu Textfeldern, Checkboxes etc. noch nicht über die Funktion `RenderControl($obj)` realisiert werden.
 - Wird der Dienst in größerem Umfang genutzt, wäre es sinnvoll, bei Auflistungen wie etwa der Suche die Anzeige im Sinne der Übersichtlichkeit auf eine Maximalanzahl von Ergebnissen pro Seite einzuschränken. Dann könnte der Nutzer bei Bedarf die Ergebnisseiten durchblättern.
 - Es ist zu prüfen, ob an manchen Stellen anstatt einer kompletten Hinweisseite (wie etwa *MfzInseratUpdateAchtung.asp* oder den Seiten zum Löschen eines Eintrags) auch eine Art JavaScript Pop-Up ausreichen würde. Dabei sollte auch hier das Corporate Design der Hochschulwebsite ausschlaggebend sein.

Anhang

Anhang A

Quelltexte

Alle in den Kapiteln 3.2 und 4 beschriebenen eigens erstellten/modifizierten Dateien (Programmseiten/Bibliotheken/Klassen) befinden sich auf der beigefügten CD-ROM.

Anhang B

Diese Referenz dient als Zusammenfassung der grundlegenden Funktionsweise des Dienstes und stellt somit eine Art einfache Gebrauchsanweisung dar.

Kurzreferenz

Dieser Dienst zur Anbahnung von Mitfahrgelegenheiten ist ein über das Studentenportal zentral zugängliches Serviceangebot der Hochschulwebsite. Als Bestandteil des Intranetbereiches wird die für einen Dienst dieser Art obligatorische Authentifizierung über den regulären Login bewerkstelligt. Nach der Anmeldung sind dem Nutzer über die Startseite des Mitfahrdienstes alle Hauptfunktionen zugänglich. Bevor jedoch selbst Eintragungen vorgenommen werden können, muss jeder Nutzer sein persönliches Profil mit den geforderten Pflichtangaben vervollständigen.

- **Grundprinzip:**

Ist eine Person auf der Suche nach einer Mitfahrgelegenheit oder möchte jemand freie Kapazitäten an Sitzplätzen in seinem Fahrzeug zur Verfügung stellen, dann kann über diesen Dienst ein Inserat erstellt werden. Dieses ist dann für andere Nutzer online - somit orts- und zeitunabhängig - zugänglich, wobei eine Suchfunktion eine detaillierte Recherche ermöglicht. Wurde etwas Passendes gefunden, kann der interessierte Nutzer bei eventuellen Rückfragen oder zusätzlich benötigten Informationen Kontakt mit dem Inserenten aufnehmen oder das Inserat direkt buchen, das heißt Sitzplätze fest reservieren. Nach stattgefundener Fahrt ist eine gegenseitige Bewertung möglich, wodurch anderen Nutzern letztendlich ein gewisses Maß an Sicherheit geboten werden soll, da am Dienst beteiligte Personen sich meist gegenseitig nicht kennen. Insgesamt handelt es sich hierbei also um eine Weiterentwicklung eines Dienstes mit reiner Anzeigefunktion – die Interaktion zwischen den Nutzern bringt ein Plus an Funktionalität und Komfort mit sich.

- **Ablauf:**

Ist bei einer Person das Fahrzeug noch nicht ausgelastet, und die freien Sitzplätze sollen zur Verfügung gestellt werden, dann wird ein Angebot eingetragen. Dieses ist dann über die Liste aller Angebote oder der Suchfunktion für andere Nutzer zugänglich. Bei positivem Interesse besteht dann die Möglichkeit entweder den betreffenden Eintrag erst einmal vorzumerken, Kontakt mit dem Inserenten aufzunehmen, oder direkt zu buchen. Bei letzterem wird für eine bestimmte Strecke eine beliebige Anzahl aus den noch offenen Plätzen reserviert.

Liegt kein passendes Angebot vor, so erstellt der Nutzer ein Gesuch. Fahrer finden so entweder für ihre eigene Strecke schnell einen Mitfahrer oder werden darauf aufmerksam und stellen die geforderte Anzahl an Plätzen bereit. Auch Gesuche sind zur Buchung vorgesehen, aber im Gegensatz zu Angeboten müssen diese im Ist-Zustand gebucht werden, das heißt Strecke und Sitzplatzanzahl sind nicht frei wählbar.

Sowohl die Inserate selbst, als auch die Buchungen können nach der erstmaligen Eintragung bei Bedarf geändert bzw. gelöscht werden - sind also nicht endgültig. Diese Verwaltungsoptionen sind über die jeweiligen personengebundenen Übersichtsseiten zugänglich.

Nach der Fahrt bzw. dem erstmalig möglichen Fahrtzeitpunkt bei regelmäßigen Inseraten können sich Fahrer und Mitfahrer gegenseitig beurteilen, indem sie für die Kriterien „Zuverlässigkeit“ und „Auftreten“ jeweils eine Note von 1 (schlecht) bis 5 (sehr gut) vergeben. Es besteht außerdem die Möglichkeit, einen kurzen einzeiligen Kommentar zur Begründung der Bewertung zu verfassen.

Alle Eintragungen – Inserate, Buchungen, Bewertungen, persönliche Profile – sind jederzeit einsehbar, entweder über die Suchfunktion oder die entsprechende Verwaltungsübersicht (Buchungsdetails sind private, alle anderen Einträge können von jedem Nutzer eingesehen werden).

Literaturverzeichnis

- [A] Matthias Kannengiesser: PHP 5 / MySQL 4, 2004, Franzis Verlag, ISBN 3-7723-7005-5
- [B] Rolf D. Stoll; Gudrun Anna Leierer: PHP 4 + MySQL, 2. überarbeitete Auflage 2000, DATA BECKER GmbH & Co. KG, ISBN 3-8158-2043-X
- [C] PHP Dokumentation
[\[http://de3.php.net/manual/de/index.php\]](http://de3.php.net/manual/de/index.php), August 2007
- [D] MySQL Dokumentation
[\[http://dev.mysql.com/doc/refman/4.1/en/index.html\]](http://dev.mysql.com/doc/refman/4.1/en/index.html), Januar 2009
- [E] SQL Tutorial
[\[http://sql.1keydata.com/de/\]](http://sql.1keydata.com/de/), August 2007
- [F] Serverseitige und Clientseitige Weiterleitungen
[\[http://blog.antikoerperchen.de/beitrag/38/gute-weiterleitung-boese-weiterleitung-redirect-per-meta-tag-javascript-htaccess-und-php.html\]](http://blog.antikoerperchen.de/beitrag/38/gute-weiterleitung-boese-weiterleitung-redirect-per-meta-tag-javascript-htaccess-und-php.html), Oktober 2007, erstellt am 13.09.2005
- [G] JavaScript / Objektreferenz / document
[\[http://de.selfhtml.org/javascript/objekte/document.htm\]](http://de.selfhtml.org/javascript/objekte/document.htm), September 2007
- [H] Firefox (ungewollter Doppelaufruf) [\[http://www.elmar-natter.de/2009/06/firefox-und-der-verflixte-doppelte-formlaufuf/\]](http://www.elmar-natter.de/2009/06/firefox-und-der-verflixte-doppelte-formlaufuf/), Juni 2009, erstellt am 03.06.2009
- [I] Browserweiche 1 [\[http://www.flexib.de/php-browserweiche-als-funktion/\]](http://www.flexib.de/php-browserweiche-als-funktion/), März 2009, erstellt am 09.02.2009
- [J] Browserweiche 2 [\[http://www.01-scripts.de/board/thread.php?threadid=144\]](http://www.01-scripts.de/board/thread.php?threadid=144), März 2009, erstellt am 14.07.2004
- [K] Gerstenberger, Sandro: Konzeption und Implementierung eines Mitfahrdienstes, Hochschule Mittweida (FH), 2003
- [L] Prof. Dr.-Ing. habil. Winkler, Lutz: Lehrveranstaltungsunterlagen Webprogrammierung, Hochschule Mittweida (FH)

Ausgewertete Mitfahrdienste

- [A] Mitfahrgelegenheit
[<http://www.mitfahrgelegenheit.de>], Juni 2007 (Auswertung), April 2008 (Screenshot)
- [B] Drive2Day
[<http://www.drive2day.de>], Juni 2007 (Auswertung), April 2008 (Screenshot)
- [C] raumobil
[<http://www.raumobil.de>], Juni 2007 (Auswertung), Mai 2008 (Screenshot)
- [D] Mitfahrzentrale
[<http://www.mitfahrzentrale.de>], Juni 2007 (Auswertung), April 2008 (Screenshot)
- [E] MFZ
[<http://www.mfz.de>], Juni 2007 (Auswertung), April 2008 (Screenshot)
- [F] Liftpool
[<http://www.liftpool.de>], Juni 2007 (Auswertung), April 2008 (Screenshot)

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 30.11.2010

Jana Bedlewski